



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE
PUEBLA**

**FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN INGENIERÍA ELECTRÓNICA, OPCIÓN
INSTRUMENTACIÓN ELECTRÓNICA**

**TESIS PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA ELECTRÓNICA:**

**SISTEMA DE APRENDIZAJE AUTOMÁTICO PARA
LA DETECCIÓN DE NEUMONÍA**

Presenta:

Lic. Edgar Ibis Tacuapan Moctezuma*

Asesores:

M.C. José Francisco Portillo Robledo

Dr. Salvador Eugenio Ayala Raggi

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca que me fue otorgada para realizar mis estudios de maestría.

A la Benemérita Universidad de Puebla y en especial a la Facultad de Ciencias de la Electrónica por darme la oportunidad de realizar mis estudios en sus instalaciones.

A la planta académica del programa de Maestría en Ingeniería Electrónica por el conocimiento que me transmitieron y por el apoyo que me brindaron.

A los miembros del jurado revisor a el M.C. Ricardo Álvarez González, a la M.C. Ana María Rodríguez y a la M.C. Selene Edith Maya Rueda por las correcciones y aportaciones realizadas al presente trabajo, por los consejos brindados y por el tiempo invertido.

A mis asesores el M.C. Francisco Portillo Robledo y al Dr. Salvador Eugenio Ayala Raggi por la asistencia, por las contribuciones e invaluable consejos a lo largo del desarrollo de la presente tesis, por transmitir sus conocimientos y por brindar motivación constante.

Al M.C. Nicolás Quiroz Hernández por impulsarme a continuar con mis estudios de maestría y por ser un ejemplo que seguir.

A mi familia por su apoyo incondicional y en particular a mis padres por haberme permitido seleccionar mi camino.

A mis compañeros y amigos con los que tuve la oportunidad de convivir y compartir experiencias.

Índice general

Resumen	1
1. Introducción	3
1.1. Anatomía del Pulmón	5
1.2. Radiografía de Tórax	8
1.2.1. Consolidación del espacio aéreo	10
1.2.2. Consolidación nodular	11
1.2.3. Consolidación intersticial	12
1.2.4. Redonda	12
1.2.5. Neumonía atípica	13
1.2.6. Absceso pulmonar	14
1.3. Trabajo Relacionado	15
1.3.1. Preprocesamiento	15
1.3.2. Conjuntos de imágenes	18
1.3.3. Métodos basados en aprendizaje profundo	19
1.3.4. Otros métodos	20
1.3.5. Detección de textura	20
1.3.6. Evaluación de desempeño	22
1.4. Objetivos	23
1.5. Justificación	24
1.6. Descripción y Diagrama de Bloques del Sistema	25
1.7. Organización de la Tesis	27
2. Marco Teórico	30
2.1. Inteligencia Artificial	30
2.1.1. Aprendizaje automático	32
2.2. Ecualización del Histograma	36
2.3. Eigenfaces	42

2.4. Normalización	46
2.5. Selección de Características	47
3. Desarrollo del Sistema Automático	52
3.1. Estructura General	52
3.2. k-NN Propuesto	54
3.2.1. Algoritmo k-NN propuesto	58
3.3. Interfaz Gráfica	61
4. Resultados	66
4.1. Conjunto de Datos Círculo	66
4.2. Conjunto de Datos Radiografías Rayos-X	70
4.2.1. Imágenes sin preprocesar	70
4.2.2. Imágenes recortadas y con mejora de contraste	72
4.3. Interfaz Gráfica	76
4.4. Discusión de Resultados	78
5. Conclusiones y Trabajo Futuro	82
5.1. Conclusiones	82
5.2. Trabajo Futuro	84
Apéndice A. Tablas de Reentrenamientos	85
Apéndice B. Pruebas Previas	88
Apéndice C. Publicaciones	93
Bibliografía	96

Índice de figuras

1.1. Vista anterior de pulmones.	6
1.2. Ramificaciones dentro de los pulmones.	7
1.3. Sacos alveolares y sección transversal vista con un microscopio.	8
1.4. Poros de Köhn y canales de Lambert.	8
1.5. Neumonía lobar.	11
1.6. Consolidación nodular.	12
1.7. Consolidación Intersticial.	13
1.8. Neumonía redonda.	13
1.9. Absceso Pulmonar.	14
1.10. Diagrama de bloques de la metodología.	26
1.11. Diagrama de bloques del sistema.	29
2.1. Fronteras de decisión con k-NN.	36
2.2. Imagen de referencia con sus niveles de gris.	37
2.3. Histograma de la imagen de referencia.	37
2.4. a)Función de densidad de probabilidad arbitraria. b)Transformación en donde se obtiene una función de probabilidad uniforme.	38
2.5. Histograma y función de distribución acumulada.	40
2.6. Imagen original e imagen con histograma ecualizado.	40
2.7. Histograma y función de distribución acumulada iniciales.	41
2.8. Histograma y FDA después de la ecualización.	41
2.9. Ejemplo de dirección de mayor variación en un conjunto de datos.	43
2.10. Imagen de N*N representada en forma de vector columna.	44
2.11. Matriz de vectores columna.	44
2.12. a)Mala separación entre clases. b)Separación entre clases buena. c)Excelente separación entre clases.	47
2.13. Funciones de distribución para una característica x.	48
2.14. Funciones de distribución para una característica x real.	48

2.15. Distribución de 3 pares de características (x_1, x_2)	49
2.16. Selección de características. De las m características disponibles, se seleccionan p	50
2.17. Visualización de la desviación estándar para una distribución normal.	51
3.1. Diagrama general del sistema (entrenamiento y prueba).	53
3.2. a) Una muestra que fue incorrectamente clasificada desde el principio. b) Se alcanza una correcta clasificación con el acercamiento de sus prototipos vecinos.	57
3.3. a) Una nueva muestra, que fue incorrectamente clasificada desde el principio, es alcanzada por sus prototipos vecinos para intentar una clasificación correcta de ésta. b) La muestra sigue resultando equivocadamente clasificada a pesar del acercamiento de los prototipos. Sin embargo, el proceso de acercamiento es interrumpido cuando una muestra almacenada en alguna memoria resulta incorrectamente clasificada debido al movimiento del tercer prototipo. En este caso, los prototipos acercados son regresados a sus posiciones iniciales.	57
3.4. Ventana principal Interfaz Gráfica.	62
3.5. Menú Archivo.	63
3.6. Menú Ayuda.	63
3.7. Explorador de archivo que se abre al seleccionar abrir o guardar en el menú archivo.	63
3.8. Ayuda.	63
3.9. Ejemplo de recorte de radiografías.	64
3.10. Acerca de.	64
3.11. Pantalla indicando cargar una imagen	65
3.12. Pantalla indicando hacer el recorte	65
4.1. Conjunto de muestras inicial.	67
4.2. Conjunto de muestras podado.	67
4.3. Prototipos iniciales.	67
4.4. Prototipos finales.	67
4.5. Tasa de reconocimiento k-NN tradicional utilizando todo el conjunto de entrenamiento y 200 muestras seleccionadas de manera aleatoria para diferentes valores de k	68
4.6. Comparación de reentrenamientos sin eliminar prototipos y eliminando prototipos.	69

4.7. Comparación de reentrenamientos con k constante y variando el valor de k después de cada entrenamiento.	69
4.8. Exactitud promedio aplicando k-NN con diferentes valores de k. La línea arena pertenece a los resultados utilizando 100 muestras de manera aleatoria por clase del conjunto de entrenamiento podado. La línea guinda muestra los resultados utilizando todo el conjunto de entrenamiento.	71
4.9. Exactitud promedio utilizando el algoritmo propuesto utilizando reentrenamientos y variando el valor de k en cada reentrenamiento.	71
4.10. Comparación entre resultados utilizando imágenes recortadas: sin mejora de contraste, con mejora de contraste utilizando CLAHE, con mejora de contraste utilizando ecualización del histograma.	72
4.11. Imágenes originales recortadas.	73
4.12. Imágenes con CLAHE.	73
4.13. Imágenes con ecualización del histograma.	74
4.14. Desempeño del algoritmo propuesto durante el entrenamiento.	75
4.15. Resultados de reentrenamientos incrementando el valor de k en 2.	76
4.16. Ejemplo 1 imagen a procesar.	77
4.17. Ejemplo 1 imagen recortada.	77
4.18. Ejemplo 1 resultado sin neumonía.	78
4.19. Ejemplo 2 imagen a procesar.	78
4.20. Ejemplo 2 imagen recortada.	79
4.21. Ejemplo 2 resultado con neumonía.	79
4.22. Variación de las imágenes de entrenamiento. En el eje X aparecen el número de eigenvalores ordenados de mayor a menor, en el eje Y se muestra la variación. El corte ocurre cuando se alcanza el 98 % de la variación total.	80
4.23. Tiempos de ejecución.	81

Índice de Tablas

1.1. Principales causas de muerte en México.	4
1.2. Métodos basados en aprendizaje profundo.	20
1.3. Soluciones basadas en otros métodos.	21
2.1. Mapeo de los valores de r_k a s_k	39
4.1. Exactitud promedio máxima utilizando 200 muestras, todo el conjunto de entrenamiento y 200 prototipos generados por el algoritmo propuesto me- diante 10 reentrenamientos.	72
4.2. Exactitud máxima para las imágenes originales, imágenes con CLAHE e imágenes con histograma ecualizado.	74
4.3. Exactitud promedio máxima utilizando todo el conjunto de entrenamiento con k-NN tradicional, reentrenamientos incrementando el valor de k en 1 y reentrenamientos incrementando el valor de k en 2 con algoritmo propuesto.	76
A.1. Reentrenamiento con el algoritmo propuesto e imágenes originales recortadas.	86
A.2. Reentrenamiento con el algoritmo propuesto e imágenes recortadas y CLAHE aplicado.	86
A.3. Reentrenamiento con el algoritmo propuesto e imágenes recortadas con ecualización del histograma.	87
A.4. Reentrenamientos incrementando el valor de k en 2, con el algoritmo pro- puesto e imágenes recortadas con ecualización del histograma.	87
B.1. Comparación de la tasa de reconocimiento obtenida con las imágenes origi- nales y con los pesos PCA	90
B.2. Resultados de reentrenamiento variando el valor de k	90
B.3. Resultados de reentrenamiento variando el valor de k y mezclando las mues- tras después de cada entrenamiento	91

B.4. Resultados de reentrenamiento variando el valor de k , con datos sin normalizar y normalizados con 49 características seleccionadas	91
B.5. Comparación de 49 características seleccionadas contra las primeras 49 características otorgadas por PCA. Ambos conjuntos de datos se normalizaron.	92

Resumen

En este trabajo se presenta una herramienta para la detección de neumonía en radiografías de rayos-X, utilizando aprendizaje automático, esta herramienta es un software que está programada totalmente en Python utilizando el entorno de desarrollo Spyder de Anaconda.

Para realizar este proyecto, se descargaron bancos de imágenes de radiografías de rayos-X a las cuales se les aplica un preprocesamiento para estandarizar las imágenes, este consiste en la mejora de contraste aplicando ecualización del histograma, además las imágenes son recortadas de tal manera que se obtenga únicamente el área de los pulmones y se redimensionan a un tamaño de 256x256, posteriormente se define un conjunto de prueba y un conjunto de entrenamiento etiquetados, sobre este último, se aplica análisis de componentes principales (PCA), los pesos resultantes de la proyección de las imágenes sobre los componentes principales se normalizan y se seleccionan las características que mejor separan las dos clases.

Estas características, se utilizan para entrenar al algoritmo de generación/selección de prototipos para que represente de mejor manera el conjunto de datos de entrenamiento, esto al entregar solo los prototipos que aporten información relevante sobre la distribución del conjunto de datos, como es el caso de las muestras que se encuentran en la frontera, una vez entrenado el modelo, se realiza la proyección del conjunto de prueba, los pesos obtenidos para cada imagen de entrada son normalizados, y se toman las características que previamente fueron seleccionadas durante la etapa de entrenamiento para predecir su etiqueta de clase. Este sistema produce como resultados un estado normal o la probabilidad de presencia de neumonía.

Dentro de las aportaciones de nuestro trabajo, se contribuye con un algoritmo de aprendizaje automático de generación/selección de instancias, a partir, de muestras reales de un conjunto de entrenamiento, para aplicaciones de clasificación. Este algoritmo está basado

en k-NN (k vecinos más cercanos) y tiene como objetivo obtener un conjunto reducido de prototipos que represente de mejor manera al conjunto de datos original. Los resultados muestran que en el caso del conjunto de datos de radiografías de rayos-X, con algoritmo propuesto utilizando 200 prototipos se obtiene una mejora en la tasa de reconocimiento con respecto al k-NN tradicional ponderado que utiliza todo el conjunto de entrenamiento de 2000 muestras.

La herramienta cuenta con una interfaz creada con Tkinter para facilitar la interacción con un usuario médico. Para el programa se utilizaron las librerías numpy, openCV, os y math.

Capítulo 1

Introducción

La inteligencia artificial (IA) es un campo altamente dinámico que está en constante expansión hacia diversas áreas adaptando algoritmos de aprendizaje automático, estos a su vez han evolucionado gracias a los avances en hardware como las GPU que permiten aplicar secuencias matemáticamente intensivas en donde se analizan grandes cantidades de datos y múltiples variables mediante métodos computacionales. En este campo, el algoritmo aprende directamente del conjunto de datos sin la necesidad de una ecuación predeterminada como modelo con el fin de realizar predicciones [1]. En el sector de la medicina conocido como detección/diagnóstico asistido por computadora (CAD), la IA ha evolucionado especialmente en la valoración por imágenes con la llegada del aprendizaje profundo basado en redes neuronales artificiales (RNA) [2].

Para interpretar la información proporcionada en las imágenes, es necesario recurrir a médicos experimentados, sin embargo, es posible que estos expertos concluyan diferentes diagnósticos de acuerdo con su propia interpretación he incluso el mismo médico puede presentar diagnósticos diferentes para las mismas imágenes en diferentes momentos [3]. Con el objetivo de lograr un diagnóstico más confiable y preciso, se presenta en este trabajo una herramienta de ayuda para radiólogos, enfocada a la detección de neumonía mediante examinación de radiografías de tórax basada en aprendizaje automático.

La neumonía es una enfermedad respiratoria infecciosa que se genera en los pulmones y puede ser contagiada a través de fluidos y secreciones corporales tales como saliva y mucosidad, que se propagan por el aire al toser o estornudar, es causada por bacterias, virus u hongos. Los patógenos bacterianos más comunes encontrados son *streptococcus (pneumococcus)*, *haemophilus influenzae* y patógenos atípicos como *mycoplasma*, *chlamydophila* y *legionella*. Mientras que los organismos virales más frecuentes son *influenza* y *parainfluen-*

za, seguidos por *respiratory syncytial* y *adenovirus* [4] [5]. Las personas más propensas a contraer neumonía son los niños, las personas mayores y personas con problemas de salud serios, esto debido a que su sistema inmunológico se encuentra vulnerable. De acuerdo con *World Health Rankings* [6] con datos de la Organización mundial de la salud (OMS) del 2017, la influenza y la neumonía son la séptima causa de muerte en México con 21,511 muertes lo que representa un 3.97% del total de decesos, ocupando el lugar 135 a nivel mundial (Tabla 1.1). Los datos de 2018 muestran un incremento en el número de muertes de 21,511 a 22,441 y a nivel mundial México pasó del lugar 135 al 122.

Tabla 1.1: Principales causas de muerte en México.

Posición	Causas de muerte	Posición a nivel mundial
1	Enfermedad coronaria	126
2	Diabetes mellitus	12
3	Derrame cerebral	158
4	Enfermedad pulmonar	54
5	Enfermedad del hígado	51
6	Enfermedad de riñón	57
7	Influenza y neumonía	135
8	Violencia	19
9	Cáncer de próstata	107
10	Accidentes de tráfico	115

Existen diferentes métodos de detección de neumonía, además de los síntomas y análisis de sangre, entre los que destacan, tomografía computarizada de tórax (CT), radiografía de rayos-X de tórax (CXR) y ultrasonido. La tomografía computarizada presenta mayor sensibilidad a costa de mayores dosis de radiación en comparación con las radiografías tórax además de que tiene un costo mayor. A pesar de que estudios muestran que el diagnóstico por ultrasonido resulta ser una mejor herramienta para el diagnóstico de neumonía en comparación con la radiografía de pecho [7][8][9], gracias a la accesibilidad, a la velocidad en la que puede ser adquirida e interpretada, además de su bajo costo, la radiografía por rayos-X es la herramienta principal para la detección de neumonía si no se está seguro del diagnóstico [4][10].

Investigadores, expertos e instituciones han creado y etiquetado conjuntos de imágenes de radiografías de tórax que están disponibles para ser utilizados como referencia para proponer soluciones que ayuden en la identificación de enfermedades. Estos bancos de

imágenes de dominio público han sido validados por especialistas por lo que son un buen punto de inicio para entrenar el sistema.

A continuación, se hace un breve resumen de la anatomía del pulmón que nos ayuda a entender mejor las zonas que afecta la neumonía y a conocer la estructura pulmonar. Posteriormente se presenta información relacionada a la apariencia que puede tomar la neumonía en las radiografías de tórax.

1.1. Anatomía del Pulmón

Con el fin de poder identificar las estructuras que determinan la presencia de neumonía en el pulmón, es necesario conocer la composición de los pulmones para entender la formación de dichas estructuras.

El sistema respiratorio es uno de los más importantes, es regulado por el sistema nervioso central y es responsable de un proceso inconsciente activo las 24 horas del día, involucra diversos músculos, huesos y órganos, además, trabaja en conjunto con el sistema circulatorio. La respiración se encarga de llevar el oxígeno a los pulmones donde se difunde en la sangre y se elimina dióxido de carbono. También se llevan a cabo funciones anexas: vibración de las cuerdas vocales para producir la voz, olfato, regulación de temperatura corporal mediante la difusión de calor, regulación de la presión sanguínea y equilibrio ácido-base.

El aparato respiratorio consta en esencia del tracto respiratorio y los pulmones, el tracto respiratorio puede a su vez dividirse en parte alta y parte baja, la parte de nuestro interés es la parte baja, se conforma por los conductos que se bifurcan para llevar aire a los pulmones: tráquea, bronquios y bronquiolos. Dentro de los pulmones el aire circula a lo largo de esta ruta sin salida que termina en los alvéolos. El aire entrante se detiene en los alvéolos, los alvéolos son sacos microscópicos revestidos de una pared capilar en donde los gases se intercambian con la circulación sanguínea y luego fluyen de regreso.

Los pulmones no son simétricos, el pulmón izquierdo es más alto y angosto que el derecho debido a la inclinación del corazón, la altura del hígado del lado derecho hace que el pulmón derecho sea más corto. Los pulmones se conforman por lóbulos, el pulmón derecho cuenta con tres: superior, medio e inferior. El pulmón izquierdo consta de 2 lóbulos: superior e inferior (Fig. 1.1, extraída de [11]). Cada lóbulo cuenta con su propio bronquio,

a estas derivaciones se les llama ramificaciones o bronquios secundarios. Antes de entrar a los pulmones, se les denomina bronquios primarios. Los bronquios tienen una estructura ramificada parecida a un árbol conocida como árbol bronquial, en donde cada rama se hace cada vez más delgada, estos ramales están cubiertos de mucosa, músculo y tejido conectivo, se extiende desde el bronquio principal hasta casi 65000 bronquiolos terminales [11].

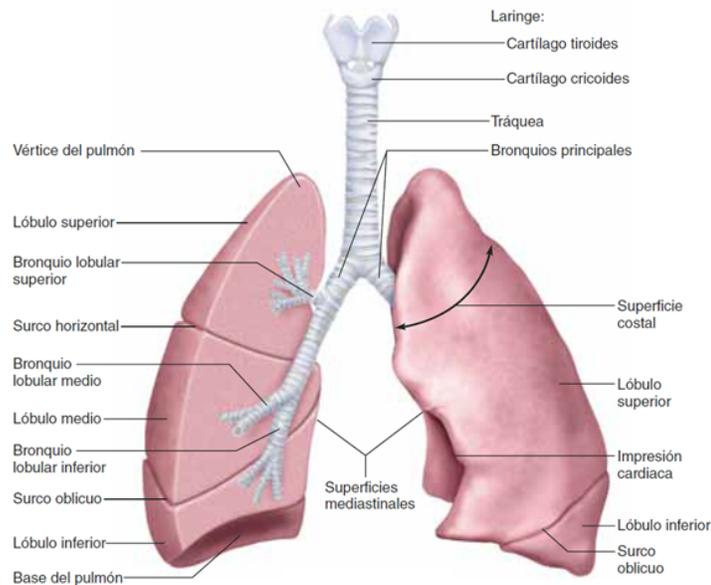


Figura 1.1: Vista anterior de pulmones.

Los bronquios lobulares o secundarios, a su vez se ramifican en bronquios segmentarios también llamados terciarios en ambos pulmones de manera dicotómica irregular. El pulmón izquierdo tiene 8 derivaciones, mientras que el derecho se divide en 10 bronquios segmentarios. Estos segmentos están separados de los segmentos adyacentes por tejido conectivo. En la Figura 1.2 (extraída de [12]) podemos observar estas ramificaciones representadas con diferentes colores.

Los bronquiolos son continuaciones de los bronquios segmentarios que suelen medir menos de 1mm. Cada bronquiolo se divide en 50 a 80 bronquiolos terminales que son las ramas finales de la división conductora. Estos miden alrededor de 0.5mm o menos de diámetro. La siguiente estructura son los bronquiolos respiratorios que tienen alvéolos brotando de sus paredes, seguidos de los conductos alveolares que finalmente terminan en los sacos alveolares, que son grupos con forma de uva (Fig.1.3a extraída de [11]). Cada pulmón humano es una masa esponjosa compuesta por 150 millones de pequeños sacos. Un alvéolo es una bolsa de 0.2 a 0.5mm de diámetro (Fig. 1.3b extraída de [11]), sus paredes elásticas permiten

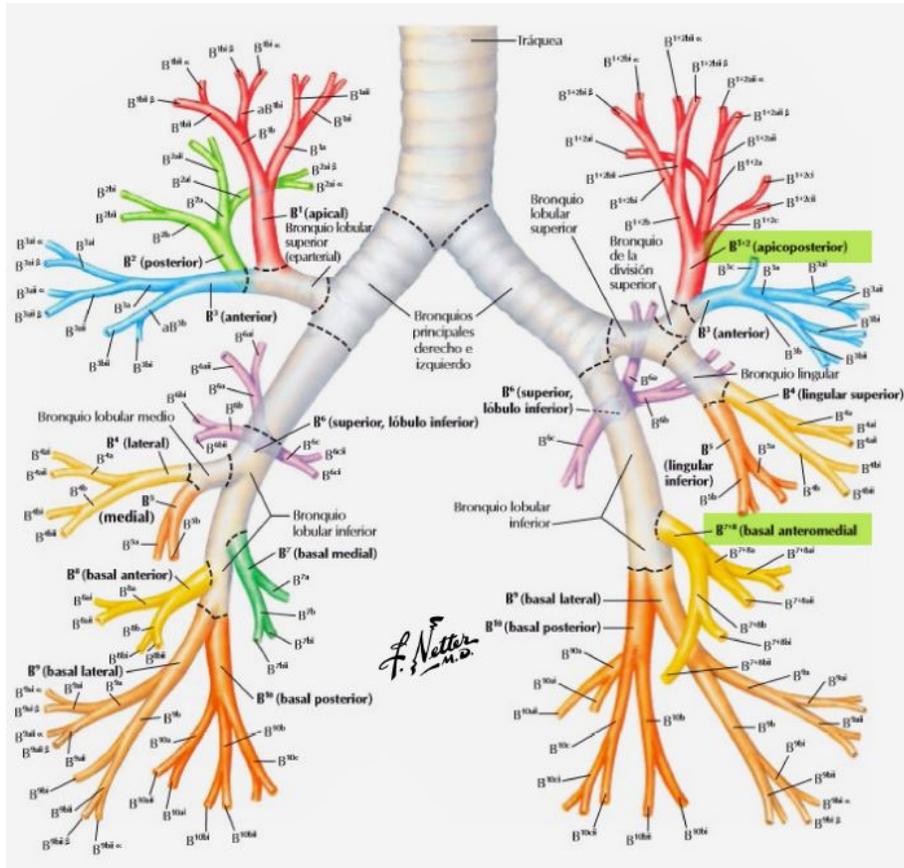


Figura 1.2: Ramificaciones dentro de los pulmones.

el flujo de aire dentro de ellas cuando se expanden y se contraen para exhalar los gases [11].

Entre alvéolos contiguos existen comunicaciones anatómicas que permiten la circulación de aire de un alvéolo a otro para que exista una distribución homogénea del aire, se denominan poros de *Köhn* o conexiones intraalveolares. Consisten en interrupciones de la pared alveolar de unas 6-12 μm de diámetro, a través de los cuales puede existir ventilación colateral que permite la llegada de aire a unidades alveolares en las que los bronquiolos que las suplen están obstruidos. A través de estos poros pueden pasar, además de los gases, líquidos y bacterias por lo que son un mecanismo de propagación de infecciones como la neumonía.

Otras vías accesorias que conectan distintas secciones de la vía aérea entre sí son los canales de Lambert (Fig.1.4 extraída de [13]). Estos canales consisten en estructuras tubulares tapizadas por epitelio que comunican los espacios alveolares con los bronquiolos terminales, e incluso con vías aéreas de mayores tamaño [14].

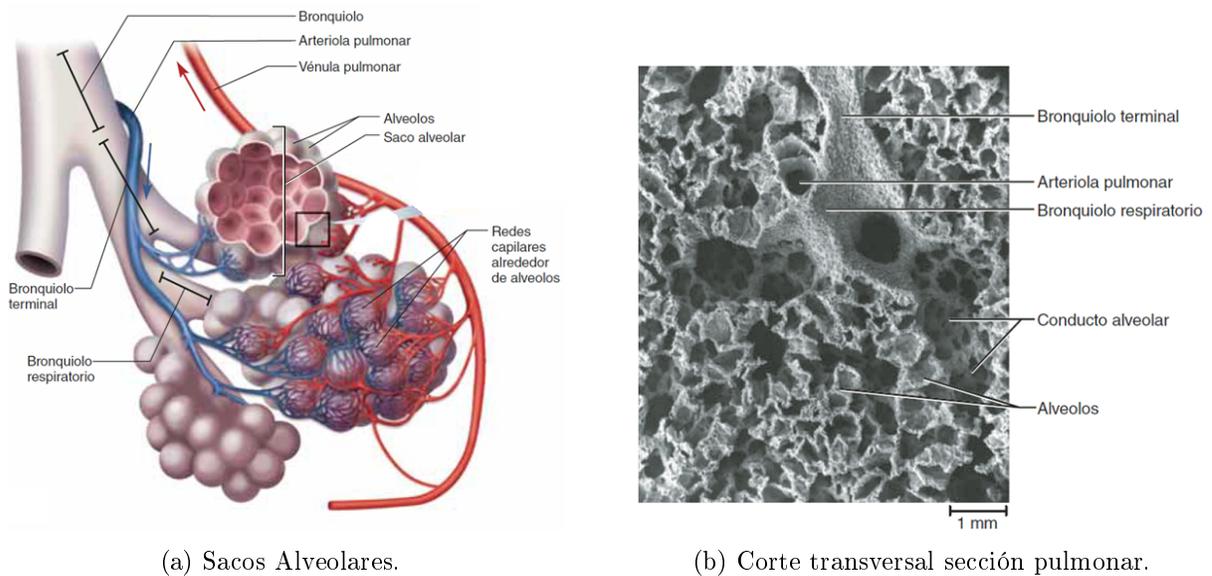


Figura 1.3: Sacos alveolares y sección transversal vista con un microscopio.

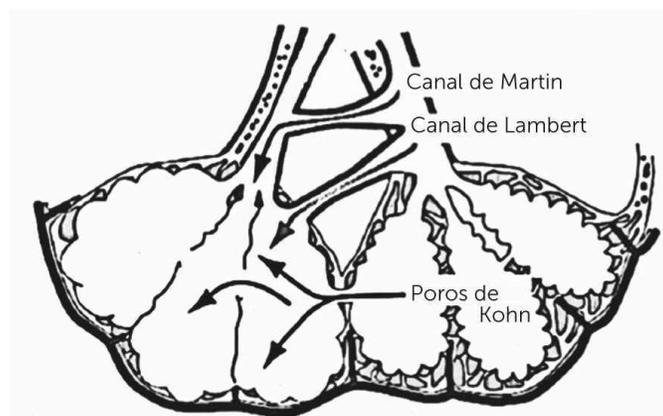


Figura 1.4: Poros de Köhn y canales de Lambert.

1.2. Radiografía de Tórax

Una radiografía es una imagen creada a partir de la aplicación de pequeñas dosis de rayos-X sobre la zona deseada, los rayos-X viajan a través del cuerpo y son absorbidos en diferentes cantidades por diferentes tejidos, las partículas de radiación atraviesan con facilidad los tejidos blandos tales como la grasa y el músculo, al igual que cavidades llenas de aire como los pulmones y son absorbidos por tejidos más densos como es el caso de huesos, corazón y vasos sanguíneos, por lo que estas imágenes son utilizadas para observar y detectar anomalías en las partes internas del cuerpo [15]. La radiografía de tórax postero anterior (PA) y la radiografía lateral siguen siendo la base de la radiología torácica. Se

realizan en máxima inspiración y con la respiración totalmente suspendida. Hay diversos procedimientos para obtener radiografías digitales de tórax, desde la digitalización de una radiografía convencional analógica a la conversión directa de la radiación transmitida de analógica a digital. Las principales ventajas de la radiografía digital son la mayor resolución, mayor contraste y la capacidad de transmitir las imágenes a un monitor para poder manipular sus características [16].

La distribución en el cuerpo de las cuatro densidades radiológicas: gas, agua, grasa y calcio o densidad metálica, determina lo que se distingue en la radiografía de tórax. Por ejemplo, no es visible la sangre en el interior del corazón, ni el corazón dentro del saco pericárdico, porque todas estas estructuras tienen la misma densidad. Este hecho explica que cualquier delimitación de una imagen radiológica dependa en parte de las diferencias en la densidad radiográfica de las diferentes estructuras vecinas[16]. En una radiografía podemos observar el corazón, los pulmones, las vías respiratorias, huesos y algunos vasos sanguíneos. Además, se pueden observar lesiones o anomalías mediante la presencia de elementos nuevos o modificación de densidad, forma, tamaño y posición sobre la región que comprende el tórax.

La neumonía es una enfermedad infecciosa que produce consolidación. La consolidación pulmonar es una afección en la cual las vías respiratorias se llenan de fluidos como agua, sangre y pus o con algún material sólido. En el caso de la neumonía, los alvéolos se llenan con pus y se inflaman, esto produce un cambio de densidad que puede ser apreciado en las radiografías de rayos-X como opacidades en forma de nubes o manchas con diferente distribución y densidad alrededor del pulmón.

Los alvéolos son cavidades semiesféricas que se encuentran al final de las ramificaciones de los bronquiolos en donde se realiza el intercambio de oxígeno con la sangre, en personas enfermas los alvéolos llenos de líquido ocasionan que la respiración se torne dolorosa, asimismo, se limita la absorción de oxígeno [5]. La detección de neumonía por medio de radiografías debe de ser diagnosticada por un radiólogo, así que la precisión del diagnóstico dependerá de la habilidades, experiencia y capacidades de este, además de la calidad de la radiografía [17]. Estos elementos hacen que la detección de neumonía sea un proceso difícil de realizar.

La clasificación de la neumonía con base en la dispersión y la apariencia radiográfica es la siguiente:

- Consolidación de espacio aéreo. (Tipo lobular o bronconeumonía)
- Consolidación nodular
- Consolidación intersticial
- Redonda
- Patrón de neumonía atípica
- Absceso pulmonar

1.2.1. Consolidación del espacio aéreo

Los espacios aéreos de los pulmones son las partes que contienen aire, excluyendo a las vías aéreas exclusivamente conductoras, comprenden los alvéolos, los conductos alveolares, y bronquiolos respiratorios. La infección que se origina en los espacios aéreos produce “neumonía lobular” (Fig. 1.5 flechas negras, extraída de [18]). Este patrón de consolidación se caracteriza por: un foco periférico solitario de opacidad densa homogénea debido a la confluencia de acinos consolidados, preservación de las vías aéreas más grandes para producir “broncogramas de aire” (bronquios llenos de aire rodeados de opacidad), propagación rápida y progresiva a áreas adyacentes y eventualmente hacia el hilio por los poros de Köhn y los canales de Lambert, los márgenes suelen estar mal definidos y borrosos, son imprecisos a menos que estén delimitados por una cisura, tienden a la coalescencia con el tejido sano que los rodea y consolidación en uno o más lóbulos del pulmón.

La neumonía Lobular se asocia clásicamente con la neumonía adquirida en la comunidad, y se debe más comúnmente a *Streptococcus pneumoniae*. Este tipo de neumonía puede afectar a uno o varios segmentos del mismo lóbulo o de otros lóbulos comprometiéndolos de manera parcial o completa, lo más frecuente es que se concentre en un solo lóbulo y no tienen un límite nítido. La tuberculosis (TB) también puede causar una consolidación homogénea similar a la neumonía lobular.

La infección que se origina en las vías aéreas y luego se propaga a los espacios aéreos produce un patrón de consolidación ligeramente diferente, denominado “bronconeumonía” (Fig. 1.5 flechas blancas). Por lo general se observa aire de manera anormal en los bronquios en el centro de la opacidad. Otras características incluyen: múltiples áreas de consolidación en parches, a menudo falta bilateral de broncogramas aéreos, coalescencia progresiva de la consolidación en parches con el tiempo. La coalescencia es la posibilidad de que dos o más materiales se unan en un único cuerpo. Se llama broncograma aéreo cuando los bronquios

se hacen visibles en una radiografía. Este fenómeno se explica por la desaparición del aire que se encuentra normalmente en los alvéolos. Este aire se ha sustituido por células o por líquido. La bronconeumonía generalmente se asocia con organismos adquiridos en el hospital (nosocomiales) como *Escherichia coli* y *Pseudomonas aeruginosa*. La bronconeumonía adquirida en la comunidad ocurre y es causada clásicamente por *Staphylococcus aureus* [18].

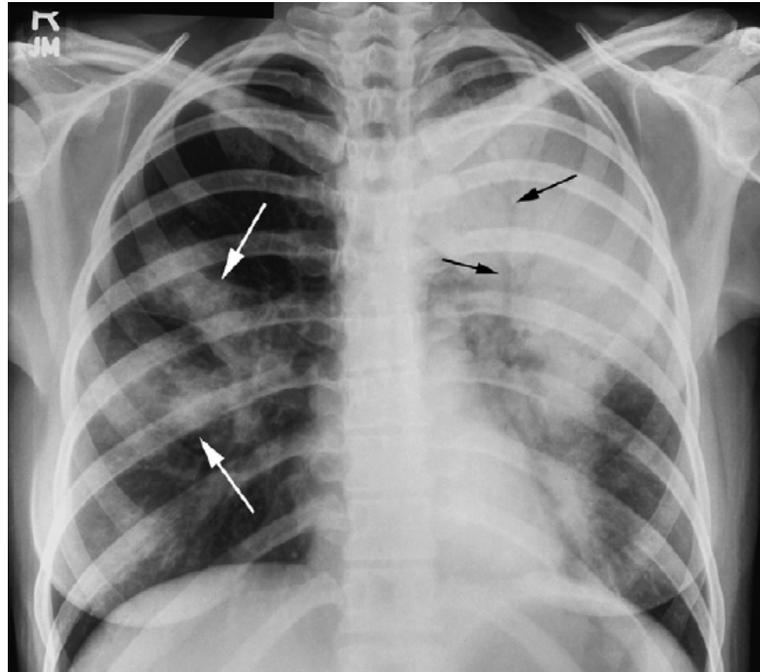


Figura 1.5: Neumonía lobar.

1.2.2. Consolidación nodular

Los nódulos son focos pequeños y redondeados de opacidad del espacio aéreo, y generalmente están asociados con infecciones no-bacterianas o bacterianas poco comunes. Los ejemplos incluyen: *Varicela zoster* (virus de la varicela), que produce neumonía nodular bilateral (Fig. 1.6 extraída de [18]), *Mycoplasma pneumoniae* (bacteria atípica), que generalmente causa neumonía nodular en un pulmón. Estos nódulos pueden fusionarse a medida que la infección progresa y producen un parche patrón tipo bronconeumonía [18].

La tuberculosis también produce comúnmente opacificación nodular, aunque predominantemente dentro de las porciones superiores de los pulmones. La progresión de la infección también puede conducir a la fusión de los nódulos y a un patrón irregular de tipo bronconeumonía [18].



Figura 1.6: Consolidación nodular.

1.2.3. Consolidación intersticial

El tejido alrededor de los sacos de aire se denomina intersticio. El intersticio pulmonar está dividido en tres compartimientos conectados entre sí por lo que las enfermedades se dispersan fácilmente de uno a otro. Es difícil notar los intersticios en radiografías simples. En personas con enfermedad pulmonar intersticial, este tejido se vuelve rígido o cicatrizado, y los sacos de aire no pueden expandirse tanto. Los lóbulos están separados por tabiques interlobulares, y el espacio entre lóbulos individuales se llama espacio intersticial. Las infecciones respiratorias no suelen producir opacificación del espacio intersticial, pero cuando lo hacen, las apariencias son bastante distintas de la opacificación del espacio aéreo: numerosas líneas cortas y opacas (reticulaciones) distribuidas simétricamente en ambos pulmones, asociación frecuente con nódulos opacos pequeños. Los organismos que producen consolidación intersticial generalmente están asociados con pacientes inmunocomprometidos, incluidos aquellos con SIDA. *Pneumocystis carinii* es un buen ejemplo (Fig. 1.7) [18].

1.2.4. Redonda

Es un tipo raro de neumonía que afecta principalmente a niños menores de 8 años cuyos poros de Köhn y canales de Lambert aún no se han desarrollado completamente o existe un defecto del desarrollo de este tejido conectivo, la consolidación presenta una

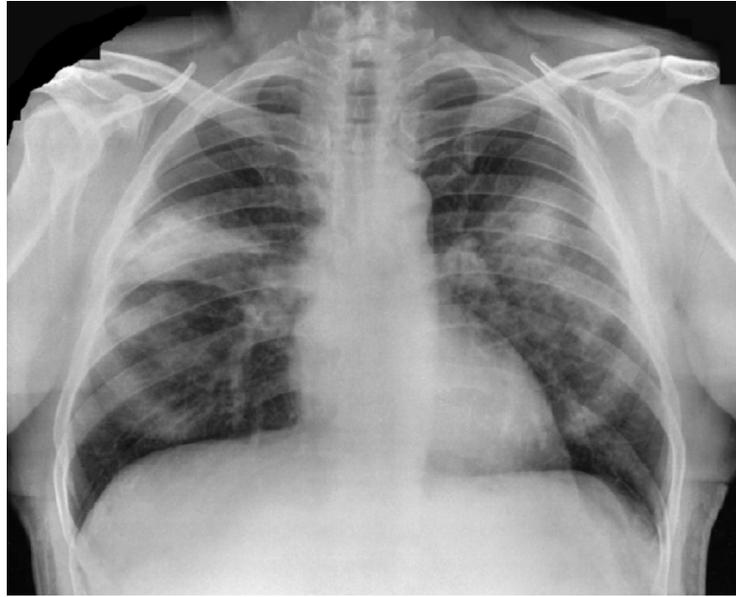


Figura 1.7: Consolidación Intersticial.

forma redonda u oval (Fig. 1.8 extraída de [19]), generalmente se observa un nódulo o masa pulmonar solitaria en los lóbulos inferiores y márgenes pleurales con dimensión promedio es de 3.8 cm de diámetro [19].

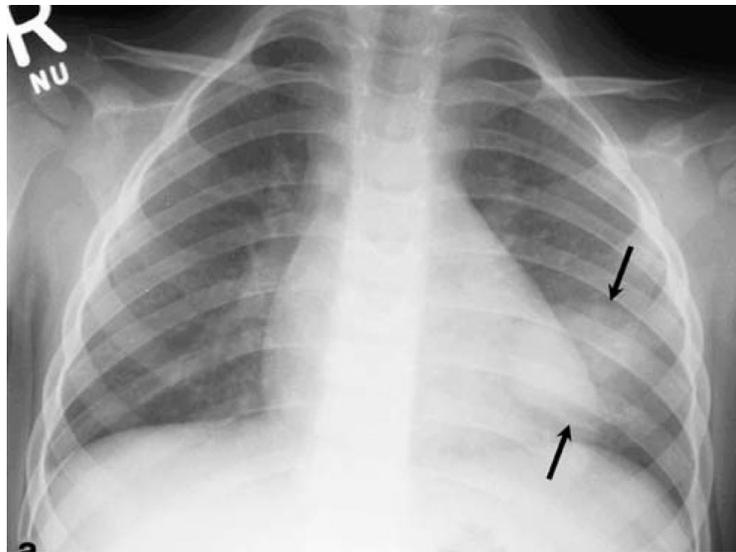


Figura 1.8: Neumonía redonda.

1.2.5. Neumonía atípica

Un término no específico que se usa para describir: neumonías causadas por bacterias o virus poco comunes, pero con patrones en radiografías comunes (lobulares o bronconeu-

mónicos), como infección por Legionela (que puede presentarse con síntomas gastrointestinales y neurológicos), neumonías con patrones inusuales en la radiografía de tórax [18]. El término atípica no se refiere a que no sea común y tiene que ver más con la causa, tiene alta morbilidad y mortalidad.

1.2.6. Absceso pulmonar

Un absceso pulmonar es una cavidad llena de pus en el pulmón, rodeada de tejido inflamado, producida por una infección. Esta es una observación histopatológica y no es perceptible en una radiografía, lo que resulta en la mayoría de los abscesos pulmonares que parecen idénticos a las áreas focales de consolidación. Las bacterias anaerobias, que normalmente se encuentran como comensales dentro de la orofaringe y el tracto gastrointestinal, pueden causar la formación de abscesos. Las bacterias ingresan a los pulmones al aspirar secreciones o vómitos, lo que lleva a lo que se denomina “neumonía por aspiración”. Esto generalmente se manifiesta como una consolidación irregular en las zonas medias de los pulmones, con o sin cavitación (Fig. 1.9 extraída de [18]). *S. aureus* y *Klebsiella pneumoniae* son dos ejemplos más de bacterias que causan cavitación dentro de las áreas de consolidación, generalmente como parte de una bronconeumonía generalizada. La TB frecuentemente también causa cavitación, predominantemente dentro de las opacidades de la zona superior y media mencionadas anteriormente [18].

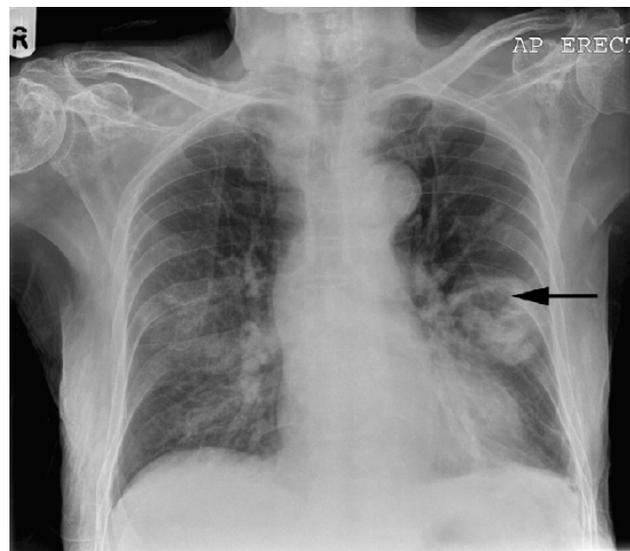


Figura 1.9: Absceso Pulmonar.

1.3. Trabajo Relacionado

En esta sección se presentan diversas soluciones para la detección de neumonía por medio de aprendizaje automático. Se presentan como primer punto técnicas de preprocesamiento utilizadas para mejorar o estandarizar ciertas características de las imágenes de entrada. Posteriormente se dan a conocer métodos basados en aprendizaje profundo para la solución de esta problemática y por último otros métodos basados en diferentes técnicas. Desafortunadamente no se puede realizar una comparación ecuánime entre las soluciones mencionadas debido a que: los bancos de datos no son los mismos, el número de imágenes de entrenamiento y prueba son diferentes, además de que no todos reportan sus resultados bajo el mismo régimen.

1.3.1. Preprocesamiento

Usualmente los conjuntos de datos se encuentran en una variedad de formatos y la mayoría de los bancos de imágenes no están estandarizados, es decir, debido a que las radiografías no fueron adquiridas con los mismos equipos ni en los mismos lugares, presentan diferencias tales como: resolución, tamaño, formato, posición, contraste además de que pueden contener ruido. Se requiere entonces aplicar algoritmos que normalicen las imágenes de entrada. El procesamiento requiere conocimiento especializado de procesamiento de imágenes, diferentes tipos de datos requieren diferentes técnicas de preprocesamiento. Las manipulaciones realizadas tienen como objetivo que el resultado de la manipulación sea más útil que los píxeles originales.

La calidad del preprocesamiento podría afectar o beneficiar el desempeño del sistema, imágenes procesadas que presenten mejoras a simple vista para el ojo humano podrían ser o no benéficas para los siguientes procesos del modelo. Las técnicas de preprocesamiento comúnmente aplicadas para imágenes de tórax son: mejora del contraste, filtrado, afilado de bordes, segmentación de la región de interés e incluso supresión de los huesos en la zona del pulmón.

Un correcto preprocesamiento de las imágenes de entrenamiento puede ayudar a mejorar los resultados de validación de los modelos como reportan en [20]. Las técnicas utilizadas para la mejora de las imágenes dentro de la bibliografía se enuncian a continuación:

Filtrado

Los filtros son una parte esencial para el procesamiento de imágenes ya que reducen la información no deseada que se pudiera encontrar en la imagen como el ruido y suavizan la imagen al reducir la variación de intensidad entre píxeles vecinos. Los filtros más utilizados para procesar imágenes de radiografías se mencionan brevemente en seguida.

Filtro de la mediana.

Consiste en promediar los píxeles vecinos del píxel central para sustituir el valor real con el nuevo promedio, se realiza una convolución sobre la imagen con una máscara de $N \times N$, donde N es el número de píxeles por lado de la máscara de convolución, generalmente se usa un valor de $N = 3$, es un filtro que preserva los bordes. Se utiliza en [21].

Filtros bilaterales.

Este filtro se aplica en [20] y en [22]. Es un filtro que suaviza las imágenes y que a diferencia de los filtros tradicionales que difuminan los bordes, éste los preserva. Combina los valores de niveles de gris con los píxeles vecinos basándose tanto en su proximidad geométrica como en su similitud fotométrica [23].

Ecualización del histograma

Los algoritmos de ecualización generan una nueva imagen a partir de aplicar una transformación que entregue un histograma con una distribución uniforme dentro de todo el rango posible para aumentar el contraste y poder observar con más detalle características en la imagen. Los algoritmos más utilizados son:

Ecualización del histograma.

Es un método automático de mejora de contraste que reparte de manera uniforme los valores de intensidad dentro del histograma. Aplicando este algoritmo y dependiendo de la distribución de intensidad de la imagen, se pueden observar detalles que anteriormente no era posible detectar. Esta técnica es utilizada en [24].

CLAHE.

Para mejorar y homogeneizar el contraste, el método más utilizado por los autores en la bibliografía revisada es la ecualización de histograma adaptativa limitada por contraste (CLAHE por sus siglas en inglés), es utilizado en [20], [21], [22], y [25]. Con este proceso la imagen se divide en bloques pequeños que se ecualizan por separado por lo que se limitan

solo a la región que corresponde a su división, para evitar la amplificación de ruido presente en la imagen se utiliza la limitación de contraste. Para eliminar posibles bordes generados por las divisiones y el hecho de que la ecualización de cada bloque es independiente de la imagen total, después de aplicar la ecualización se aplica interpolación bilineal [26]. En [20] mencionan que utilizan este método porque evita el realce excesivo del contraste y la amplificación del ruido en las imágenes.

Corrección gamma

La corrección gamma sirve para codificar o decodificar la luminancia de imágenes a tonos apropiados para ser interpretados por el ojo humano, esto es debido a que las cámaras no perciben la luz de la misma manera que la visión humana lo hace. Esta corrección se utiliza para que las imágenes no se vean tan oscuras en monitores al compensar ciertas propiedades del ojo humano. Esta técnica es utilizada en [21].

Otras técnicas

En [27] realizan una mejora del contraste que consiste en obtener el promedio de todos los píxeles de la imagen y restarlo a cada píxel para posteriormente dividir el resultado entre la desviación estándar.

Otsu thresholding. Es un método automático de umbralización que es capaz de separar los píxeles en dos clases, primer plano y fondo con lo que se obtiene una imagen binaria. En [24] utilizan este umbral para segregar la parte sana del pulmón de la región afectada por la neumonía.

Artículos como [28–30] reportan mejoras al seccionar el área de los pulmones.

Data augmentation

Aunque esta técnica no es parte del preprocesamiento de imágenes, se menciona porque es muy utilizada en redes neuronales para balancear las clases. Consiste en generar nuevas imágenes a partir de otras de la misma clase aplicando técnicas de rotación, redimensión, traslación, adición de ruido, filtrado, entre otras. Las imágenes resultantes son ligeramente diferentes pero iguales en esencia. Esto ayuda a que la red no memorice una imagen y se produzca un sobre ajuste, además ayuda a aumentar la tasa de reconocimiento como se puede constatar en [31].

1.3.2. Conjuntos de imágenes

A continuación, se presentan las descripciones de los bancos de datos revisados en la bibliografía y que se encuentran disponibles para descarga, es decir se omiten los bancos de datos privados. Estos bancos fueron recopilados por diversos especialistas, instituciones y hospitales de diferentes partes del mundo.

ChestX-ray8

Banco de datos con 108948 imágenes de rayos-X de 32717 pacientes únicos, el número 8 se refiere a la cantidad de enfermedades que incluye y con las que están etiquetadas las imágenes. Las enfermedades son: *Atelactasis*, *Cardiomegaly*, *Effusion*, *Infiltration*, *Mass*, *Nodule*, *Pneumonia*, *Pneumothorax* y algunas imágenes pueden tener etiquetas múltiples. Para crear estas etiquetas, los autores utilizaron el procesamiento del lenguaje natural para clasificar texto de las clasificaciones de enfermedades de los informes radiológicos asociados, banco de datos de *The National Institute of Health of America*. La precisión en el etiquetado es mayor al 90 %. El tamaño de las imágenes es de 1024 x 1024 [32].

ChestX-ray14

Es una versión mejorada de ChestX-ray8 que agrega más imágenes y enfermedades. Las enfermedades agregadas son: *Consolidation*, *Edema*, *Emphysema*, *Fibrosis*, *Pleural Thickening*, *Hernia* de igual manera pueden contener múltiples etiquetas. Las imágenes fueron extraídas de la base de datos de *National Institutes of Health Clinical Center*. Consiste en 112120 imágenes frontales de 30805 pacientes etiquetadas mediante procesamiento del lenguaje natural con datos extraídos de los informes radiológicos asociados [32].

Banco de datos Mendeley

Este conjunto contiene miles de imágenes validadas de OCT y radiografías de tórax analizadas en [33], en ese artículo mencionan que recolectaron y etiquetaron un total de 5232 imágenes de rayos-X de tórax de niños, incluidas 3883 caracterizadas por representar neumonía (2538 bacterias y 1345 virales) y 1349 normales, de un total de 5856 pacientes para entrenar el sistema de IA. Las imágenes se dividen en un conjunto de entrenamiento y un conjunto de pruebas de pacientes independientes. Las imágenes se etiquetan como [enfermedad - identificación aleatoria del paciente - número de imagen] [34].

Las imágenes de rayos-X del tórax (anterior-posterior) se seleccionaron de pacientes pediátricos de uno a cinco años del Centro Médico para Mujeres y Niños de Guangzhou,

Guangzhou. Todas las imágenes de rayos-X del tórax se realizaron como parte de la atención clínica de rutina de los pacientes. Para el análisis de imágenes de rayos-X de tórax, todas las radiografías de tórax se examinaron inicialmente para el control de calidad eliminando todos los escaneos de baja calidad o ilegibles.

JSRT database

La base de datos fue creada por *Japanese Society of Radiological Technology (JSRT)* en cooperación con *Japanese Radiological Society (JRS)* en 1998. Desde entonces, la base de datos JSRT ha sido utilizada por varios investigadores en el mundo para diversos fines de investigación, como procesamiento de imágenes, compresión de imágenes, evaluación de visualización de imágenes, diagnóstico asistido por computadora (CAD), sistema de archivo y comunicación de imágenes (PACS), para capacitación y pruebas. Las imágenes de esta base de datos son las que tienen mejor resolución (2048 x 2048). Poseen amplio rango de densidad (12 bits, escala de grises 4096). El formato de imagen es universal (sin encabezado, datos brutos) además contienen la edad del paciente, sexo, diagnóstico (maligno o benigno), coordenadas X e Y del nódulo, diagrama simple de ubicación del nódulo, grado de sutileza en la detección visual de nódulos [35].

COVID-19 radiography database

Un equipo de investigadores de la Universidad de Qatar y la Universidad de Dhaka, Bangladesh, junto con sus colaboradores de Pakistán y Malasia, en colaboración con médicos, crearon esta base de datos de imágenes de rayos-X de tórax para casos positivos de COVID-19 junto con Imágenes de neumonía normal y viral. En este banco hay 1200 imágenes positivas para COVID-19, 1341 imágenes normales y 1345 imágenes de neumonía viral. Es una base de datos pública disponible en Kaggle que combina varias bases de datos públicas y también recopilando imágenes de artículos publicados relacionados a COVID-19 [36].

1.3.3. Métodos basados en aprendizaje profundo

Se han presentado varias soluciones en años recientes para la detección de neumonía de manera automática por medio de radiografías que en su mayoría utilizan métodos basados en *Deep Learning*. En la tabla 1.2 se muestran algunos de los artículos revisados junto con datos importantes, se puede observar que los resultados se reportan utilizando diferentes métricas. Todas ellas utilizan redes neuronales convolucionales y la mayoría de ellas utiliza transferencia del conocimiento, lo que significa que toman redes ya hechas y entrenadas

para solucionar una problemática distinta pero relacionada con la nueva problemática y solo hacen un reentrenamiento con un conjunto pequeño de imágenes en comparación con el número de imágenes utilizadas en el primer entrenamiento. También es notable que hay trabajos con una exactitud cercana al 99 %.

Tabla 1.2: Métodos basados en aprendizaje profundo.

Autores	Año	Modelo	Método	Resultados
Wang et al. [28]	2019	CNN	CNN Cooperatives	24.09 Score
Rakshit et al. [27]	2019	CNN	Transfer Learning	77.88 AUC.
Sirazitdinov et al. [37]	2019	CNN	Transfer Learning	77.50 F1-Score
Togaçar et al. [31]	2019	CNN	mRMR M.L. classifiers	99.41 ACC.
Liang et al. [38]	2019	CNN	Transfer Learning	97.6 Recall rate
Chouhan et al. [39]	2019	CNN	Transfer Learning	96.4 ACC.
Islam et al. [40]	2019	CNN	Transfer Learning	98.99 ACC.
Acharya et al. [41]	2019	CNN	Deep Siamese Network	0.95 AUC
Chhikara et al. [21]	2020	CNN	Transfer Learning	90.1 ACC.
Sarcar et al. [20]	2020	CNN	Residual Learning	99.72 AUC., 99.28 ACC.
Rahman et al. [42]	2020	CNN	Transfer Learning	98 ACC.
Yao et al. [43]	2020	CNN	YOLOv3	46.84 Precision
Labhane et al. [44]	2020	CNN	Transfer Learning	97 ACC.

1.3.4. Otros métodos

En la Tabla 1.3, se muestran métodos que no utilizan redes neuronales convolucionales para extraer características, reportan una exactitud (ACC) más baja en comparación con las CNN. También es notable que el número de trabajos es menor ya que el aprendizaje profundo está en auge y la mayoría de los investigadores está trabajando bajo este paradigma.

1.3.5. Detección de textura

El trabajo relacionado, muestra que es posible determinar la presencia de neumonía en imágenes de radiografías por medio de detección de textura. Diferentes técnicas de detección de textura han sido aplicadas. En [48] y [47] emplean características Haralick que se basan en la matriz de coocurrencia de la cual extraen 14 características, en [48] definen como las más relevantes de estas la varianza, la suma promedio y la suma-varianza. En [45], utilizan dos conjuntos de características de los cuales uno se basa en texturas de la

Tabla 1.3: Soluciones basadas en otros métodos.

Autores	Año	Modelo	Método	ACC.
Xie et al. [45]	2006	Funciones basadas en momentos y matrices de coocurrencia	Características de textura	70.94 %
Noor et al. [46]	2014	Discriminación estadística, wavelet texture and modPC	Texturas Wavelet	70 %
Sharma et al. [24]	2017	Otsu thresholding	Procesamiento de imágenes	N.D.
Varela et al. [47]	2019	NN	Texturas Haralick	93 %
Vamsha et al. [48]	2019	Reglas de decisión	Características de Textura	86 %
Chandra et al. [29]	2020	Perceptrón Multicapa	Regresión logística y clasificación vía regresión	95.63 %
Khatri et al. [30]	2020	Earth Mover's Distance	Comparación de histogramas	83.33 %
Ambita et al. [49]	2020	Support Vector Machines	Locally Adaptive Regression Kernels	96 %

matriz de coocurrencia, calculan la energía, la entropía e inercia.

En [46] y [50] utilizan 12 características wavelet en cuatro direcciones, las características son las siguientes: energía promedio, entropía, contraste, homogeneidad, desviación estándar de valor, desviación estándar de energía, coeficiente máximo wavelet, coeficiente mínimo wavelet, valor máximo de energía, máximo vector suma de energía, máximo vector columna de energía y promedio de números de cruce por cero. Promedio, varianza, desviación estándar, falta de simetría, curtosis, suavidad, uniformidad, entropía son técnicas usadas en [29].

En [51] se aplican tres conjuntos de características, el primero basado en histogramas, el segundo es un vector de gran dimensión que incluye características de momento de intensidad, borde, textura y forma, utiliza el descriptor de Tamura, CEDD, FCTH, Hu, MPGE-7, CLD, EHD, Gabor, etc., el conjunto 3 es un conjunto de características proporcionadas por regioncrops de Matlab. En [49] utilizan kernels adaptativos de regresión (LARK por sus siglas en inglés) que capturan estructuras locales. En [52] se propone el uso de filtros multiescala para la detección de tuberculosis que son filtros de derivativos gaussianos y son los coeficientes de una expansión de Taylor de la imagen.

1.3.6. Evaluación de desempeño

Las métricas de evaluación de desempeño más comunes de clasificación se muestran a continuación. La matriz de confusión no es propiamente una métrica de desempeño sino más bien es una forma de visualización del comportamiento del sistema. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo las diferentes clases o resultados de la clasificación.

		Resultado de la predicción		
		Positivo	Negativo	
Valor Actual	Positivo	VP	FN	VP+FN
	Negativo	FP	VN	FP+VN

La exactitud se refiere al porcentaje total de elementos clasificados correctamente. Se considera la manera más directa de la calidad de los sistemas de clasificación. El valor generalmente se proporciona en porcentaje (0-100 %), en donde entre más alto mejor. En casos donde las clases no están balanceadas, no resulta ser la mejor opción.

$$Exactitud = \frac{VP + VN}{(VP + FN) + (FP + VN)} \quad (1.1)$$

La métrica de sensibilidad muestra la cantidad de verdaderos positivos que el modelo ha clasificado en función del número total de valores positivos. También se conoce como *recall*. Una sensibilidad alta significa pocos falsos negativos.

$$Sensibilidad = \frac{VP}{VP + FN} \quad (1.2)$$

La especificidad es la fracción de verdaderos negativos, indica la capacidad de nuestro estimador para discriminar los casos positivos, de los negativos. Se trata de los casos negativos que el algoritmo ha clasificado correctamente.

$$Especificidad = \frac{VN}{FP + VN} \quad (1.3)$$

La precisión representa el número de verdaderos positivos que son realmente positivos en comparación con el número total de valores positivos predichos. Una precisión alta significa pocos falsos positivos.

$$Precisión = \frac{VP}{VN + FP} \quad (1.4)$$

F1-score es la combinación de las métricas de precisión y exhaustividad. La mejor puntuación F1 es igual a 1 y la peor a 0.

$$F1Score = \frac{2 * VP}{2VP + FN + FP} \quad (1.5)$$

En donde:

Verdaderos Positivos (VP): cuando la clase real es Verdadero y la predicción Verdadero.

Verdaderos Negativos (VN): cuando la clase real es Falso y el pronosticado es Falso.

Falsos Positivos (FP): cuando la clase real es Falso y el pronosticado es Verdadero.

Falsos Negativos (FN): cuando la clase real es Verdadero y el valor predicho es Falso.

1.4. Objetivos

Objetivo General

Desarrollar un sistema basado en aprendizaje automático que determine, a través de análisis de imágenes de radiografías de tórax, la presencia de neumonía.

Objetivos Específicos

1. Investigar y valorar diferentes métodos de detección de neumonía en el estado del arte y elaborar una tabla con las características de los diferentes métodos.
2. Obtener un conjunto adecuado de imágenes de radiografías de Tórax.
3. Analizar y determinar la forma, escala y posición de las estructuras dentro de los pulmones que determinan un estado normal y presencia de neumonía.
4. Evaluar técnicas de procesamiento y acondicionamiento de imágenes y elegir la más apropiada para acentuar las estructuras mencionadas en el objetivo anterior.
5. Determinar si es posible o no, modelar dichas estructuras utilizando técnicas conocidas para detección de textura.

6. De acuerdo con el tipo de estructuras evaluar técnicas de extracción de características tales como HOGs, Gabor, etc., que proporcionen una mejor separación de las dos clases que los píxeles originales.
7. Evaluar métodos básicos de clasificación y métodos basados en aprendizaje profundo para mejorar el desempeño en la detección de neumonía.
8. Programar en Python todos los algoritmos que integren un sistema de detección de neumonía el cual cuente con una interfaz de fácil manejo para un usuario médico.

1.5. Justificación

Siendo una de las principales causas de decesos a nivel mundial y debido a que la neumonía debe de ser tratada en las primeras etapas para reducir el riesgo de muerte, es indispensable proporcionar un diagnóstico oportuno y conciso por parte de los especialistas. Una de las maneras más viables y económicas para detectar la neumonía es mediante la examinación de radiografías de Tórax, el problema de estas radiografías es que se necesita un especialista altamente capacitado para poder detectar con precisión la presencia de neumonía en el paciente.

La falta de capital humano capacitado, los tiempos largos necesarios para capacitar personal, el error humano inherente, la uniformidad de la experiencia por parte de los expertos, la calidad de las radiografías, son algunos de los elementos que influyen en la correcta evaluación de las enfermedades por medio del análisis de radiografías y hacen necesario crear herramientas que soporten el diagnóstico proporcionado por el especialista y reduzcan los tiempos de estudio.

La evolución de la inteligencia artificial generada en gran parte gracias al avance exponencial de las capacidades de procesamiento permite proponer soluciones competentes en el ámbito médico con ayuda de los vastos bancos de imágenes que han sido proporcionados por diferentes organizaciones e instituciones hospitalarias y educativas. Esto a su vez posibilita corroborar los algoritmos desarrollados para medir su confiabilidad bajo distintas circunstancias.

El Dr. Curtis Langlotz profesor de radiología e investigador en informática biomédica en el centro médico de la universidad de Stanford, en 2017 dijo que era completamente obvio que en 5 años la inteligencia artificial sería mejor que los radiólogos pero que eso

no significaba que su trabajo estuviera en riesgo, más bien aumentaría sus capacidades y haría sus vidas mejores y más fáciles, también respondió a la pregunta de que si en algún momento los radiólogos serían reemplazados por la inteligencia artificial a lo que respondió: “Yo digo que la respuesta es no, pero los radiólogos que usen Inteligencia Artificial reemplazarán a los que no la usen”[53].

El presente trabajo de tesis pretende agilizar los tiempos de examinación de radiografías de tórax mediante un algoritmo descrito en Python que realice este proceso de manera automática, además de validar la decisión del analizador y darle herramientas que le permitan competir y mejorar en la detección de neumonía. Se utiliza Python debido a que es un lenguaje de alto nivel elegante y flexible muy conocido que exige mínimos requerimientos de instalación, que además es *Open Source* por lo que constantemente se generan mejoras y librerías. Cuenta con una licencia Pública General de GNU, esto permite que el programa se pueda aprovechar sin restricciones y sin la necesidad de adquirir una licencia por parte de los radiólogos u hospitales.

1.6. Descripción y Diagrama de Bloques del Sistema

La detección de neumonía se realizará utilizando algoritmos de inteligencia artificial, aplicados a bancos de imágenes de radiografías de tórax públicos disponibles para descarga. El siguiente paso es analizar estructuras que determinan la presencia de neumonía, posteriormente se evalúan métodos de acondicionamiento de imágenes para hacer resaltar estas estructuras. Una vez analizadas las estructuras el siguiente paso es extraer características sobre las zonas de interés. Por último, se evalúan técnicas de clasificación y con base en los resultados se realizarán ajustes para mejorar la precisión. Se busca que el algoritmo una vez entrenado, sea capaz de diferenciar entre pulmones con presencia de neumonía y pulmones normales con una interfaz simple para que pueda ser fácilmente utilizada por un especialista.

La metodología se resume en el siguiente diagrama de bloques figura 1.10.

Generalmente en un sistema de aprendizaje automático se requieren aplicar los siguientes pasos:

- Preprocesamiento
- Extracción de las regiones de interés



Figura 1.10: Diagrama de bloques de la metodología.

- Extracción de características sobre las regiones de interés
- Entrenamiento del Modelo
- Clasificación

Preprocesamiento. Los bancos de imágenes contienen radiografías de diferentes hospitales por lo que no fueron tomadas con las mismas condiciones ni con los mismos aparatos por lo que pueden presentar variaciones de intensidad, contraste, tamaño o alineación por lo que puede ser necesario aplicar algoritmos para mejorar estas características. En este trabajo el preprocesamiento consiste en mejorar el contraste de las imágenes aplicando ecualización del histograma.

Extracción de las regiones de interés. Es importante segmentar las zonas sobre las cuales se van a extraer las características para reducir el número de errores ocasionados por otros elementos presentes en la imagen, además, de mejorar la exactitud del algoritmo. En este paso se podría segmentar la sección de los pulmones para posteriormente extraer características únicamente sobre esta región. En la etapa de entrenamiento esta segmentación se realiza con un recorte cuadrado que se aplica a todas las imágenes por igual de manera automática eliminando secciones de la imagen de arriba, abajo, izquierda y derecha de manera que se obtenga solamente el área de los pulmones. En la interfaz gráfica el recorte sea hace manual con el ratón seleccionando el área deseada.

Extracción de características sobre las regiones de interés. Una vez que se segmentó la zona de interés, es necesario obtener características que nos permitan diferenciar entre personas normales y personas con neumonía. La neumonía se presenta en las radiografías como opacidades en escala de grises en diferentes zonas del pulmón y con diferente textura por lo que el siguiente paso es extraer características sobre la sección del pulmón para detectar opacidades que en algunos casos no son visibles a simple vista o que incluso pueden ser síntomas de otra enfermedad como por ejemplo el cáncer de pulmón. Este

proceso se realiza mediante eigenfaces y aunque no son características como tal, los pesos producidos de la proyección son los vectores que nos proporcionaron los mejores resultados.

Entrenamiento del Modelo. Para entrenar el modelo, se divide el banco de imágenes en: imágenes de entrenamiento e imágenes de prueba. Para observar el comportamiento se utilizan herramientas para comparar la clasificación hecha por el modelo de acuerdo con las etiquetas. El algoritmo seleccionado para la clasificación es un algoritmo propuesto basado en k -NN y que utiliza el promediado parecido a k -means que ayuda a la generalización.

Clasificación. De acuerdo con las características que mejor describan la presencia de neumonía, el algoritmo debe diferenciar a que clase pertenece la imagen de entrada entre las dos posibles salidas: normal o probabilidad de presencia de neumonía.

El diagrama de bloques de la figura 1.11 describe de manera gráfica y general los pasos a seguir para implementar el algoritmo.

1.7. Organización de la Tesis

La tesis sigue el siguiente orden: el capítulo 1 se hace un breve resumen de la anatomía del pulmón que nos ayuda a entender mejor las zonas que afecta la neumonía y a conocer la estructura pulmonar. Posteriormente se presenta información relacionada a la apariencia que puede tomar la neumonía en las radiografías de tórax. En la siguiente sección, se enuncian diferentes soluciones propuestas dentro del estado del arte para esta problemática, se mencionan los objetivos específicos y generales, así como también la justificación. Por último, se presenta el diagrama de bloques del sistema propuesto.

El capítulo 2 se mencionan los conceptos relacionados con el trabajo realizado, se describe a grandes rasgos el concepto de Inteligencia Artificial, el aprendizaje automático y describe los temas que intervienen directamente con el trabajo de tesis como son: el algoritmo de k -vecinos más cercanos (k -NN), la ecualización del histograma, la normalización el algoritmo de Eigenfaces y la selección de características.

En el capítulo 3 se describe el proceso realizado para detectar la neumonía en imágenes de rayos-X, comienza con la estructura general, se describe a detalle el algoritmo propuesto para la selección/generación de prototipos y se muestra la interfaz gráfica. También, se describe el algoritmo general y se detalla la principal contribución de este trabajo, que

es un nuevo algoritmo de aprendizaje automático basado en k *Nearest Neighbors* para generar/seleccionar instancias a partir de muestras reales de un conjunto de entrenamiento para aplicaciones de clasificación.

El siguiente capítulo es el de resultados (capítulo 4). Aquí se presentan los resultados al aplicar el algoritmo k-NN propuesto a dos conjuntos de datos, un conjunto de datos sintético y con el conjunto de datos de las radiografías de rayos-X, además se muestra el uso de la interfaz gráfica.

El último capítulo es el de conclusiones (capítulo 5) en donde se menciona el porqué de los procesos realizados, la principal aportación y el trabajo futuro.

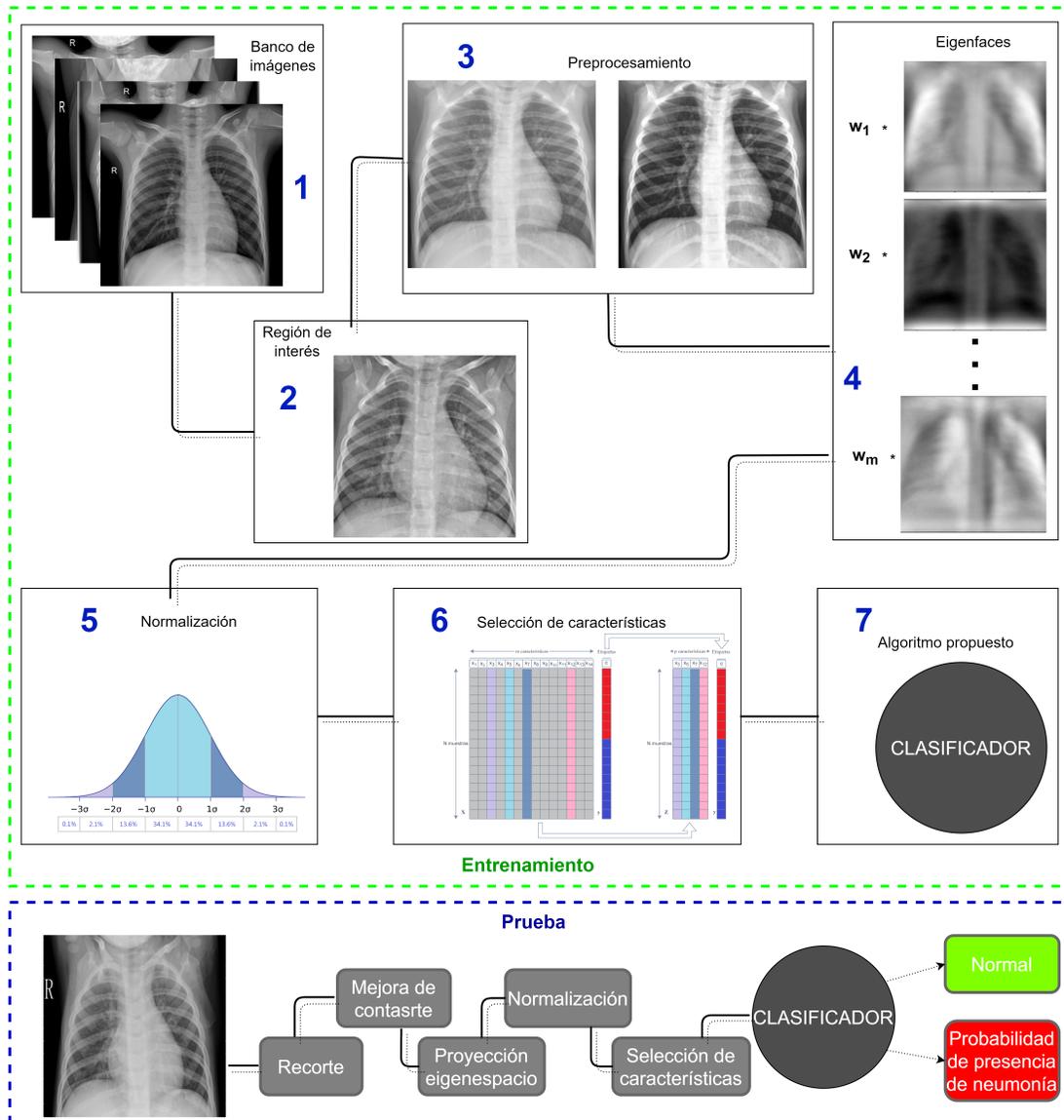


Figura 1.11: Diagrama de bloques del sistema.

Capítulo 2

Marco Teórico

En este capítulo se mencionan los conceptos relacionados con el trabajo realizado, se describe a grandes rasgos en concepto de Inteligencia Artificial, el aprendizaje automático y diferentes técnicas y algoritmos evaluados en la presente tesis.

2.1. Inteligencia Artificial

El término de inteligencia artificial fue mencionado por primera vez en los años cincuenta, sin embargo, es en los últimos años en donde se ha desarrollado con gran potencial y ha comenzado a usarse de manera cotidiana. Actualmente la inteligencia artificial se encuentra inmersa en la vida diaria, *chatbots*, anuncios y contenido en redes sociales, reconocimiento de correo spam, listas de reproducción personalizadas en plataformas de *streaming*, asistentes de voz y predictores de buscadores son las aplicaciones más comunes.

La inteligencia artificial es una rama de la ciencia computacional que combina algoritmos con el objetivo de crear máquinas que presenten comportamientos humanos considerados inteligentes, no obstante, no existe un acuerdo para una definición universal. Existen diferentes definiciones para describir la inteligencia artificial que se pueden clasificar dentro de cuatro enfoques principales: sistemas que piensan como humanos, sistemas que piensan racionalmente, sistemas que actúan como humanos y sistemas que actúan racionalmente. Los dos primeros contemplan procesos mentales y razonamiento, mientras que los dos últimos hacen referencia a la conducta. Ejemplos para cada uno de estos enfoques son: redes neuronales artificiales, robots, sistemas expertos y agentes inteligentes respectivamente. Para el enfoque de comportamiento humano se puede considerar inteligente a un sistema si es capaz de procesar el lenguaje natural para comunicarse, si puede representar el co-

nocimiento para almacenar lo que conoce o siente, si posee razonamiento automático para responder preguntas y extraer conclusiones de la información previamente almacenada y si aprende de manera automática para adaptarse a nuevos escenarios y detectar nuevos patrones [54]. Una definición más global dada por John McCarthy en 1955 fue: “*El objetivo de la IA es desarrollar máquinas que se comporten como si fueran inteligentes*”.

Estas definiciones pueden crear altas expectativas en las personas y las hacen pensar en máquinas capaces de aplicar sus habilidades para resolver cualquier problema al igual que lo hace un humano, que desafortunadamente en la actualidad la tecnología aún no es capaz de realizar. La aplicación de la inteligencia artificial busca que los sistemas puedan realizar tareas de manera autónoma y que sean capaces de tomar decisiones sobre qué hacer en un entorno de trabajo determinado bajo parámetros establecidos, es decir que están restringidas y solo operan en ambientes controlados. En estos momentos no existe un sistema completamente inteligente que pueda ser comparado con un humano, si bien es cierto que en algunas tareas se han logrado resultados que sobrepasan las capacidades humanas y superan por mucho a los humanos, aun no es posible integrar estos elementos en conjunto y no se espera lograr en un futuro próximo. Los métodos y formalismos en la IA no han sido declarados firmemente por lo que la IA no es una receta universal sino grupos de herramientas para tareas muy diferentes disponibles como bibliotecas de software [55] y esto ha dado lugar a la creación de una multitud de disciplinas que abarcan diversos campos de desarrollo, las más conocidas son:

- Visión por computadora
- Reconocimiento de patrones
- Robótica
- Razonamiento automatizado
- Procesamiento de lenguaje natural
- Aprendizaje automático

La IA más que cualquier otra ciencia es interdisciplinaria, ya que se basa en descubrimientos de campos tan diversos como las matemáticas, lógica y filosofía, la investigación de operaciones, estadística, ingeniería de control, procesamiento de imágenes, lingüística, filosofía, psicología, ciencia cognitiva y neurobiología [56].

En el presente documento solo se abordará la subárea de interés de la inteligencia artificial conocida como aprendizaje automático o aprendizaje máquina.

2.1.1. Aprendizaje automático

El aprendizaje máquina (*machine learning*) es un campo que evoluciona constantemente, el término es tan viejo como las propias computadoras. Se refiere a que una máquina sea capaz de aprender y adaptarse, mejorando sus predicciones de manera autónoma con el tiempo sin necesidad de que el programador intervenga, imitando la forma en que aprendemos los humanos, es decir que aprenda con base en la adquisición de experiencia. Debido a la naturaleza compleja de características o patrones de las tareas en donde se emplea, el programador no puede proveer de manera finamente detallada las especificaciones de como las tareas deben ser realizadas. Estas características lo hacen especialmente aplicable en donde se requieren reconocer patrones complejos en grandes cantidades de datos.

Las aplicaciones en donde se ha desarrollado especialmente son en coches autónomos, recomendaciones de canciones o películas en plataformas digitales, motores de búsqueda, finanzas, procesamiento de imágenes, visión por computadora, detección de rostros, procesamiento de lenguaje natural, detección de objetos, robótica y diagnóstico médico. Esto es posible gracias a la disponibilidad de grandes volúmenes de datos y a la capacidad de cómputo asequible de alto rendimiento.

El aprendizaje se consigue con el entrenamiento del sistema, en donde mediante datos de ejemplo y patrones, enseñamos a la computadora a que aprenda dichos patrones para identificarlos y asociarlos para posteriormente a la entrada de un nuevo dato que no se encuentra dentro del banco de datos de aprendizaje, realice predicciones. Lo que se precisa como máquina, es en realidad un algoritmo. Existen tres técnicas principales que se emplean actualmente para entrenar modelos, también existen formas híbridas como el aprendizaje semisupervisado.

- Aprendizaje no supervisado
- Aprendizaje semisupervisado
- Aprendizaje por refuerzo
- Aprendizaje supervisado

Aprendizaje no supervisado

Bajo este paradigma, se agrupan e interpretan datos basados únicamente en los datos de entrada que no se encuentran etiquetados y buscan similitudes. Aquí el sistema no tiene una respuesta correcta y el sistema se encarga de encontrar estructuras o atributos dentro

de la base de datos que permita organizarlos de alguna manera. El término en inglés se conoce como *clustering*. Los algoritmos más comunes de agrupamiento son:

- K-medias (*K-means*)
- Agrupamiento difuso (*Fuzzy C-means*)
- Agrupamiento jerárquico (*Hierarchical*)
- Modelos de mezcla gaussiana (*Gaussian Mixture*)
- Modelo oculto de Markov (*Hidden markov model*)

Aprendizaje semisupervisado

Las aplicaciones de este aprendizaje son compartidas con el aprendizaje no supervisado, variando en que, a diferencia de este último, el sistema se alimenta con datos etiquetados y datos no etiquetados. Son una alternativa para cuando no se tienen grandes cantidades de datos para realizar el entrenamiento y el costo de realizar el etiquetado es demasiado alto.

Aprendizaje por refuerzo

Este aprendizaje se produce cuando la máquina aprende por medio de prueba y error. El objetivo es que el sistema tome acciones que maximicen la recompensa en menor tiempo ante diversas situaciones. Es comúnmente usado en el área de robótica.

Aprendizaje supervisado

Es resultado de un modelo predictivo basado en datos de entrada y retroalimentado con los datos de salida, los datos de entrada contienen etiquetas de clases por lo que se conoce la respuesta deseada para cada conjunto de datos de entrada. Esto hace que el aprendizaje supervisado se utilice en aplicaciones donde datos históricos se utilizan para predecir eventos futuros.

Las técnicas utilizadas más comunes son la clasificación y la regresión. La primera predice respuestas discretas mientras que la segunda se utiliza para predecir respuestas continuas. Los algoritmos para clasificación más conocidos son:

- Maquinas de vectores de soporte (*Support Vector Machines*)
- Análisis discriminante (*Discriminant Analysis*)
- Clasificador bayesiano ingenuo (*Naive Bayes*)
- Vecinos más cercanos (*k Nearest Neighbor*)

Para la regresión, los algoritmos más utilizados son:

- Regresión lineal (*Linear Regression*)
- Regresión de vectores de soporte (*Support vector regression*)
- Proceso de regresión gaussiana (*Gaussian process regression*)
- Árboles de decisión (*Decision trees*)
- Redes Neuronales (*Neural Networks*)
- Métodos de conjunto (*Ensemble methods*)

La mayoría de las aplicaciones utilizan aprendizaje supervisado, dentro de estos algoritmos encontramos al aprendizaje profundo en donde se utilizan redes neuronales que intentan emular la percepción del cerebro humano a través de redes neuronales artificiales, conectando grandes cantidades de estas organizadas en capas, en donde cada capa aprende patrones una de la otra.

Para el desarrollo de nuestro algoritmo, utilizamos el algoritmo de *k-Nearest Neighbor*, que es un algoritmo supervisado. Este algoritmo se detalla a continuación.

K vecinos más cercanos

El algoritmo k-NN es un algoritmo de clasificación supervisado, es decir que, los datos deben de estar correctamente etiquetados. Clasifica nuevas muestras tomando en cuenta la similitud con otras muestras por cercanía. Realiza una votación para definir a que clase pertenece la muestra con base en la mayor cantidad de vecinos que lo rodean. Utiliza todo el conjunto de datos para realizar la clasificación por lo que puede ser computacionalmente intensivo y además requiere el uso de una cantidad considerable de memoria para almacenar todas las muestras.

El primer paso de este algoritmo consiste en calcular la distancia entre la nueva muestra a clasificar y todas las muestras almacenadas. Posteriormente se seleccionan los k elementos más cercanos a la muestra a clasificar, hay diferentes formas de calcular la distancia y la más común es la distancia euclidiana. La distancia euclidiana se calcula con la siguiente ecuación:

$$d(\mathbf{v}, \mathbf{x}_j) = \sqrt{\sum_{i=1}^p (v_i - x_{j_i})^2} \quad (2.1)$$

donde \mathbf{v} es el vector a clasificar, \mathbf{x}_j es la muestra j -ésima del conjunto de entrenamiento, i es la característica i -ésima de los vectores a comparar y p es el número de características

del vector.

El último paso es clasificar la muestra realizando una votación que puede ser simple o ponderada de acuerdo con la distancia, la clase con la votación ganadora definirá la clase de la muestra a clasificar. El algoritmo 1 describe los pasos a realizar para un conjunto de entrenamiento \mathbf{X} con sus respectivas etiquetas de clase \mathbf{y} con n elementos. Para mejorar la clasificación utilizando k-NN, en este trabajo se utiliza una ponderación gaussiana que consiste en otorgar mayor peso a los vecinos en relación con la distancia. Al realizar esto, los votos de los vecinos más cercanos tendrán mayor contribución. Esta función de base radial se define a continuación:

$$\omega_j = e^{-\frac{d_j^2}{2\sigma^2}} \quad (2.2)$$

donde d_j es la distancia entre la muestra a clasificar y el vecino j , y σ es un parámetro a elegir. La etiqueta de la muestra a clasificar se obtiene de la clase ganadora:

$$y_v = \arg \max_{y_c \in c} \sum_{i=1}^k w_i [y_c = y_i] \quad (2.3)$$

Algoritmo 1: kNN

Data: k , $D = \{\mathbf{X}, \mathbf{y}\}$;
 $\mathbf{x}_i \in R^p \wedge y_i \in [1, \dots, c]$
Result: Clase y_v
 Sea \mathbf{v} la muestra a clasificar;
for $j=1:n$ **do**
 \triangleright Distancia de la muestra a clasificar contra cada muestra almacenada;
 $\mathbf{d}_j(\mathbf{v}, \mathbf{x}_j)$;
end
 Elegir las k menores distancias en \mathbf{d} ;
 Realizar la votación de acuerdo con la clase;
 Elegir la votación ganadora yc ;
 $y_v \leftarrow yc$ \triangleright Asignamos etiqueta;
return y_v \triangleright Retornamos la etiqueta predicha;

La elección del valor de k es muy importante y depende de las características de los datos, un valor de k muy bajo podría provocar una frontera sobre ajustada (Fig.2.1c) mientras que un valor de k muy grande marcará una frontera sub-ajustada (Fig.2.1a). La frontera ideal depende de la distribución de los datos, en la Fig. 2.1b se muestra un ejemplo de la frontera esperada para esa distribución de datos en específico. Lo que se

busca es encontrar un valor de k que más generalice una correcta clasificación.

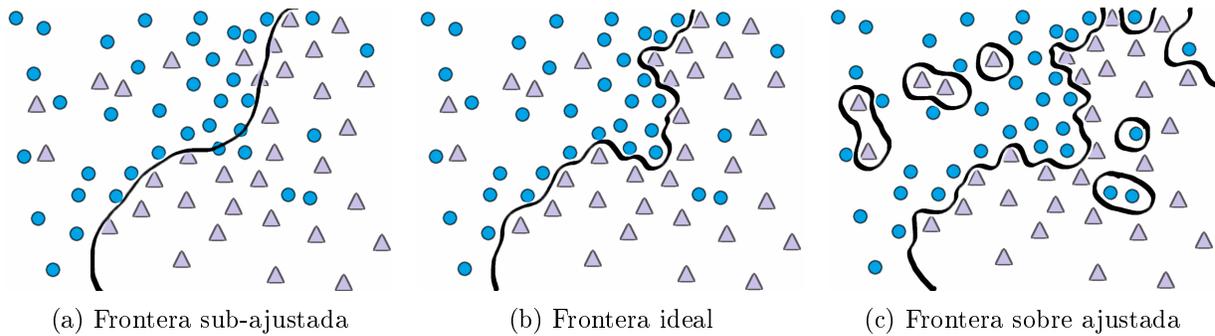


Figura 2.1: Fronteras de decisión con k -NN.

El algoritmo k -NN [57] o *k* vecinos más cercanos sigue siendo una de las técnicas de clasificación más utilizadas debido a su simplicidad y facilidad de implementación. Sin embargo, los requisitos de almacenamiento y el costo computacional aumentan a medida que aumenta el número de muestras. La razón es simple: se hace una comparación de similitud por proximidad entre la muestra a clasificar y cada una de las muestras de referencia disponibles en el conjunto de datos.

2.2. Ecuación del Histograma

En el área de reconocimiento de patrones, una imagen con poco contraste no provee mucha información. Una imagen con un contraste ideal tiene un histograma con el mismo número de frecuencias en cada barra del histograma, esto es debido a que todos los niveles de gris se encuentran distribuidos de manera uniforme. Un histograma es una representación en forma de gráfico de los niveles de intensidad de cada píxel presentes en una imagen. En cada barra del histograma se almacenan las frecuencias de los niveles de intensidad que se encuentran dentro de ese rango específico de valores. En el eje Y se muestran las frecuencias de las distintas tonalidades de gris mientras que en el eje X aparecen los niveles de intensidad que para una imagen de 8 bits es de 0 a 255 ($[0, L-1]$; $L=256$).

A continuación, en la figura 2.2a, se ejemplifica la construcción de un histograma a partir de una imagen simple, en donde el número de valores posibles L es igual a 7. La dimensión de la imagen es 5×7 por lo que contiene 35 píxeles. El histograma resultante para la imagen 2.2a, tomando sus niveles de gris (Fig.2.2b) se muestra en la figura 2.3. En el eje x se tienen los niveles de gris de la imagen y en eje y el número de píxeles que se encuentran en la imagen.

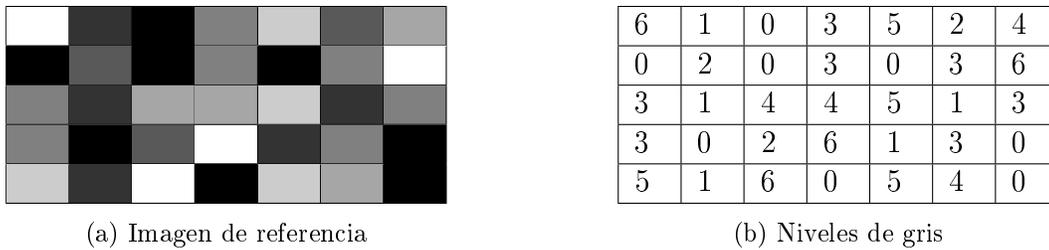


Figura 2.2: Imagen de referencia con sus niveles de gris.

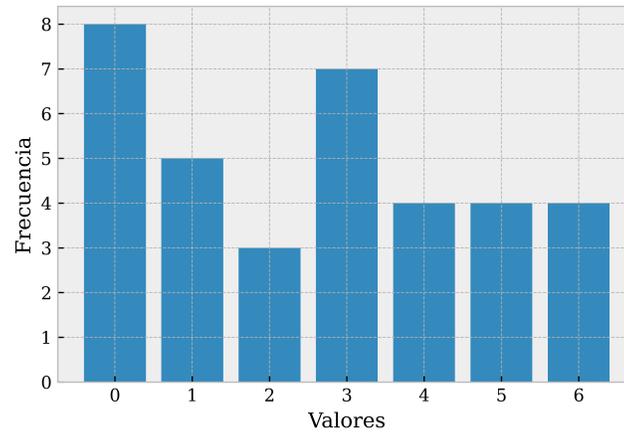


Figura 2.3: Histograma de la imagen de referencia.

En imágenes más grandes por lo general los valores del histograma no se encuentran repartidos uniformemente por lo que pueden verse con un contraste pobre. La ecualización del histograma tiene como objetivo obtener una distribución de probabilidades uniforme de los niveles de gris de una imagen. Para hacer esto, el algoritmo expande el histograma de manera uniforme de manera que los pixeles se distribuyan en todo el rango de valores posibles. De esta manera, las áreas con contraste bajo adquieren mayor contraste, por lo que en la imagen existen tanto áreas claras como áreas oscuras. Al realizar el proceso de ecualización del histograma, la función de distribución acumulativa se convierte en lineal en comparación con la original que suele ser más curva. En funciones continuas, independientemente de la forma de la función de densidad de probabilidad, es posible obtener una distribución uniforme.

Denotamos r como las intensidades una imagen a procesar. Esta variable se encuentra en el rango $[0, L-1]$, con $r=0$ para el color negro y $r=L-1$ para el blanco. Se aplica una transformación de la forma:

$$s = T(r) \quad 0 \leq r \leq L - 1 \tag{2.4}$$

que produce un valor de intensidad s , para un valor de intensidad dado r en la imagen de entrada. Se asume que $0 \leq T(r) \leq L - 1$ para $0 \leq r \leq L - 1$. La transformación inversa:

$$r = T^{-1}(s) \quad 0 \leq s \leq L - 1 \quad (2.5)$$

La intensidad de una imagen puede ser vista como una variable aleatoria en el intervalo $[0, L-1]$. Denotamos a las funciones de densidad de probabilidad (FDP) como $p_r(r)$ y $p_s(s)$ de los niveles de intensidad r y s . Partiendo de la teoría de probabilidad, si $p_r(r)$ y $T(r)$ son conocidas y $T(r)$ es continua y diferenciable sobre el rango de valores de interés, entonces la FDP de la variable transformada s puede ser obtenida como:

$$\begin{aligned} p_s(s) &= p_r(r) \left| \frac{dr}{ds} \right| \\ &= p_r(r) \left| \frac{1}{(L-1)p_r(r)} \right| \\ &= \frac{1}{L-1} \quad 0 \leq s \leq L-1 \end{aligned}$$

El resultado obtenido es una función de densidad de probabilidad uniforme y además esto será válido para cualquier $p_r(r)$ independientemente de su forma [58]. En la figura 2.4 se ilustra este concepto.

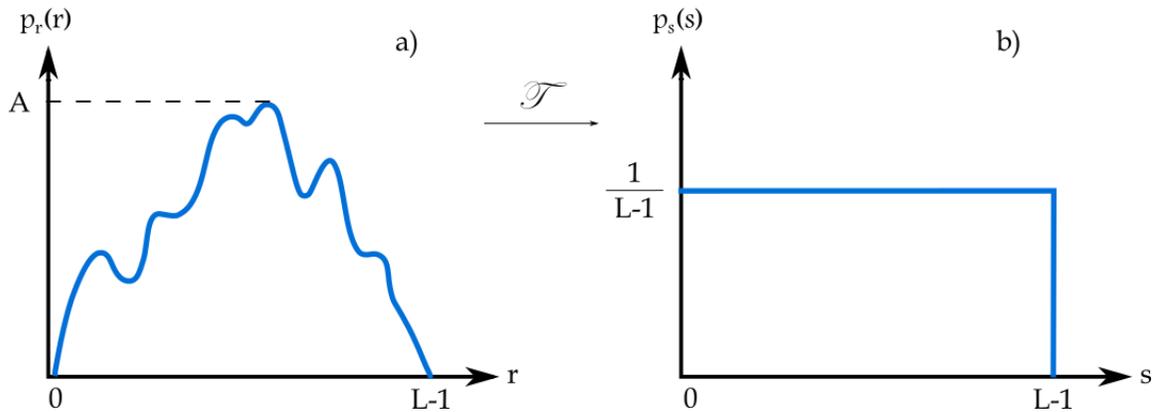


Figura 2.4: a) Función de densidad de probabilidad arbitraria. b) Transformación en donde se obtiene una función de probabilidad uniforme.

En valores discretos la probabilidad de ocurrencia de un nivel de intensidad r_k en una imagen digital se aproxima por:

$$p_r(r_k) = \frac{n_k}{MN} \quad (2.6)$$

donde MN es el número total de píxeles en la imagen y n_k es el número de píxeles que tienen una intensidad r_k . El histograma normalizado se presenta con $p_r(r_k)$ con $r_k \in [0, L - 1]$. La forma discreta de la transformación es:

$$s_k = T(r_k) = L - 1 \sum_{j=0}^k p_r(r_j) \quad k = 0, 1, 2, \dots, L - 1 \quad (2.7)$$

donde L es el número de valores posibles de intensidad en la imagen. Al proceso de transformar una imagen utilizando la ecuación 2.7, que mapea niveles de intensidad r_k a valores s_k se le conoce como ecualización del histograma o linealización del histograma.

Para ejemplificar el proceso de mapeo tomemos como ejemplo la imagen 2.2a, el valor de $L = 7$, $MN = 35$ y el rango es $[0,6]$. Tomemos el ejemplo r_1 :

$$s_1 = T(r_1) = 6 \sum_{j=0}^1 p_r(r_j) = 6(p_r(r_0) + p_r(r_1)) = 2.227$$

Este resultado junto con los demás valores los podemos ver en la tabla 2.1. Como solo se pueden tener valores enteros se hace un redondeo que se muestra en la última columna de la tabla.

Tabla 2.1: Mapeo de los valores de r_k a s_k .

r_k	n_k	$p_r(r_k) = n_k/MN$	s_k	$round(s_k)$
$r_0 = 0$	8	0.2285	1.371	1
$r_1 = 1$	5	0.1428	2.227	2
$r_2 = 2$	3	0.0857	2.742	3
$r_3 = 3$	7	0.2000	3.942	4
$r_4 = 4$	4	0.1142	4.627	5
$r_5 = 5$	4	0.1142	5.312	5
$r_6 = 6$	4	0.1142	6.000	6

La figura 2.5a muestra el histograma normalizado, de esta manera, $p_r(k)$ da una estimación de la probabilidad de ocurrencia del nivel de gris r_k , como el histograma está normalizado, la suma de sus componentes es igual a 1. En la figura 2.5b podemos observar la función de transformación.

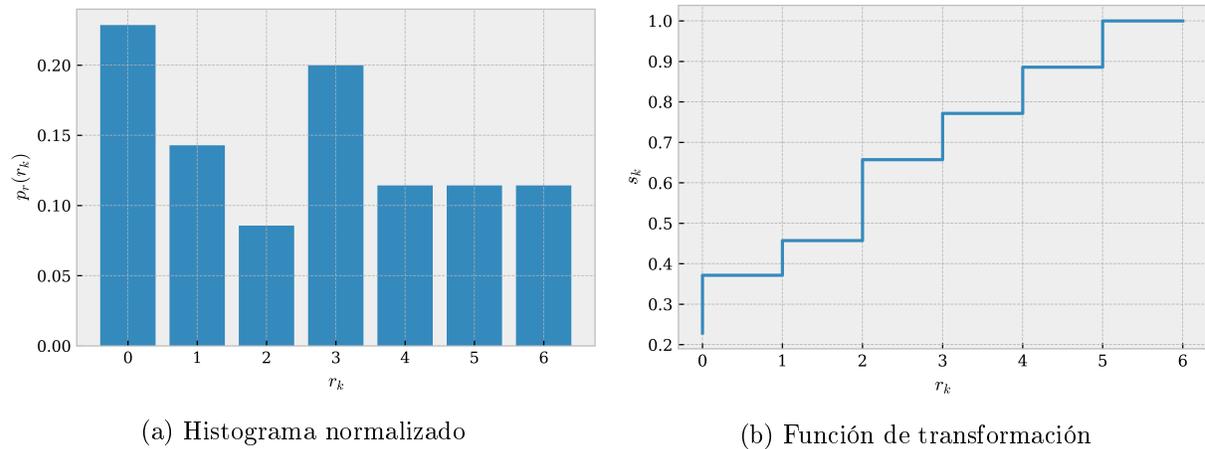


Figura 2.5: Histograma y función de distribución acumulada.

Al hacer una ecualización del histograma se busca que la imagen utilice todo el rango de valores posibles de tonos de gris y que, además, estos tonos se usen con la misma frecuencia. Como ya se había mencionado anteriormente, en funciones discretas esto no es posible por lo que en el histograma aparecerán valores de gris que no se utilizan, para ver esto de manera más detallada se muestra la imagen 2.6a junto con su histograma (Fig. 2.7a) y su función de distribución acumulada (FDA) (Fig. 2.7b). Se puede observar que la FDA no es lineal, en una imagen con un histograma ecualizado, esta función es una rampa.

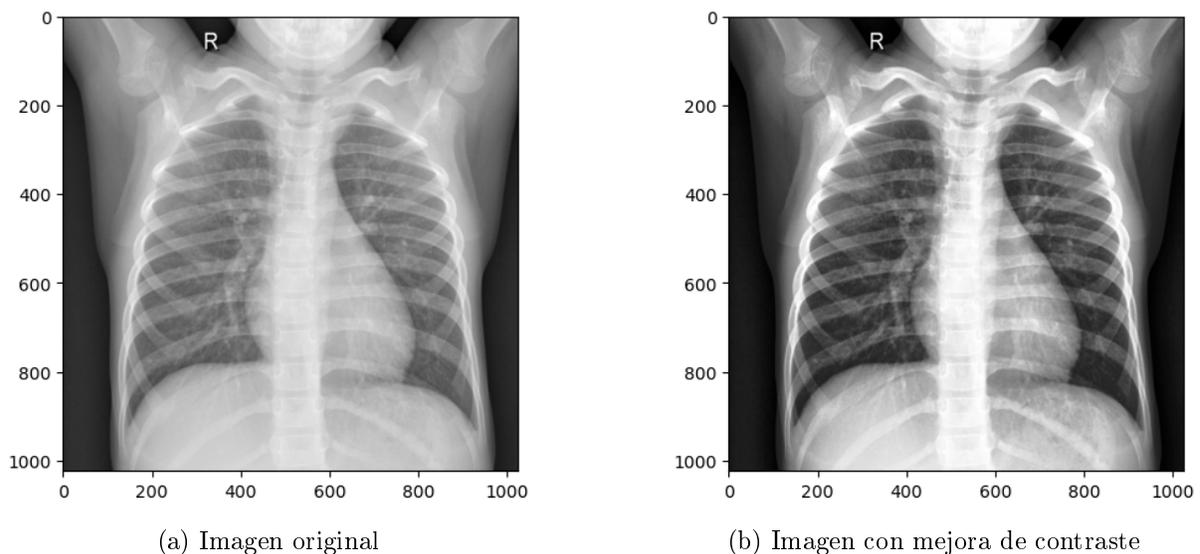


Figura 2.6: Imagen original e imagen con histograma ecualizado.

Además, se observa que hay pocos píxeles con valores cercanos a cero al igual que cercanos a 255, en el rango de 0 a 80, de la misma forma se observa poca frecuencia, esto

también se observa en la FDA con un leve incremento en este rango. Hay un pico en el rango de 180 a 210 debido a que estos valores aparecen con mayor frecuencia en la imagen.

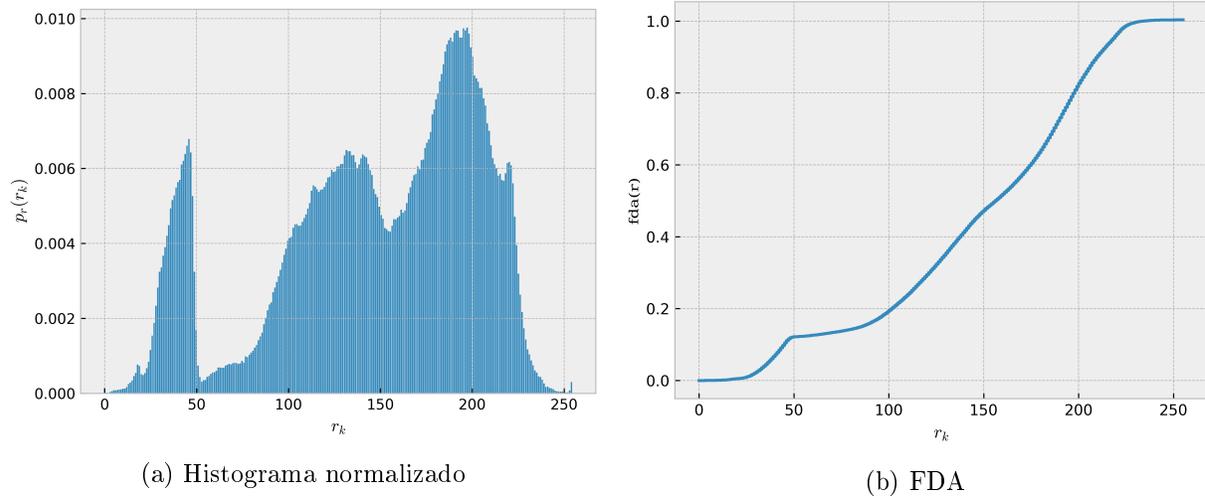


Figura 2.7: Histograma y función de distribución acumulada iniciales.

Al aplicar la ecualización de histograma a la imagen 2.6a obtenemos la imagen 2.6b, en la figura 2.8a observamos su histograma ecualizado en donde se nota que los niveles de tonos de gris están mejor distribuidos. También se observa que su función de distribución acumulada (Fig. 2.8b) se asemeja a la ideal.

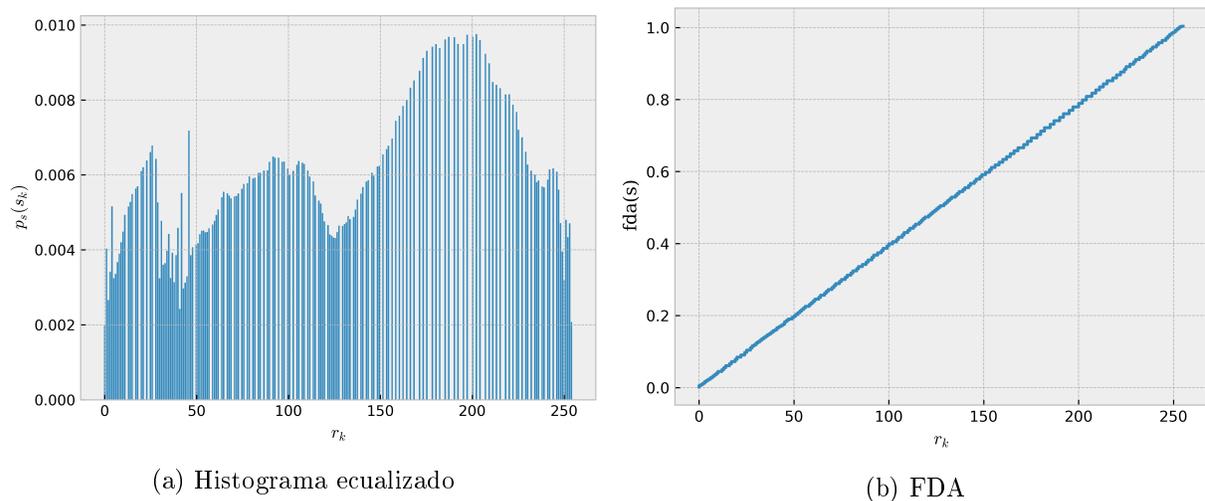


Figura 2.8: Histograma y FDA después de la ecualización.

2.3. Eigenfaces

Un paso fundamental y a la vez problemático en análisis de patrones en imágenes, es la extracción de características que ayuden a diferenciar entre las clases posibles. Existen diferentes métodos para realizar este proceso, pero no existe un método universal por lo que generalmente se recurre a realizar pruebas con diferentes métodos y elegir el más adecuado o hacer combinaciones de las mejores características de cada uno de ellos.

Eigenfaces un método basado en análisis de componentes principales (PCA por sus siglas en inglés), utilizado como compresor, tomando en cuenta los componentes principales (de mayor variación) sobre un conjunto de datos. Estos componentes son una combinación lineal de las variables originales y sintetizan la mayor parte de la información contenida en los datos, además, estos componentes son ortogonales e independientes. Por ejemplo, en la figura 2.9 observamos un conjunto de datos en donde la línea guinda representa la dirección de mayor variación que es el primer eje en un nuevo sistema de coordenadas. La segunda componente no correlacionada es un eje perpendicular a la primera. PCA es un algoritmo no supervisado, por lo que no es necesario tener el conjunto de entrenamiento etiquetado.

Las relaciones entre los vectores individuales y todo el conjunto de datos se realizan de manera estadística, especialmente se utiliza la covarianza que se emplea en vectores de varias dimensiones, con la que se busca saber si existen relaciones entre las características en las dimensiones. A los datos originales se les resta el promedio con el objetivo de estandarizar las variables y tengan una media de cero debido a que PCA identifica las direcciones con mayor varianza y si no se hace este proceso, las variables de escala mayor dominarán al resto. La covarianza se mide entre dos dimensiones, si se mide entre la misma dimensión, se obtiene la varianza. La varianza se calcula con la ecuación 2.8 y covarianza se calcula con la ecuación 2.9. Para cada muestra se multiplica la diferencia entre el valor en X menos el promedio en X por el valor en Y menos el promedio en Y, se suman todos los resultados y se dividen entre el número de muestras menos 1. Como la multiplicación es conmutativa, $cov(X, Y) = cov(Y, X)$.

$$var(X) = \frac{\sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})}{n - 1} \quad (2.8)$$

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \quad (2.9)$$

Si los valores que se obtienen al calcular la covarianza son positivos, esto indica que

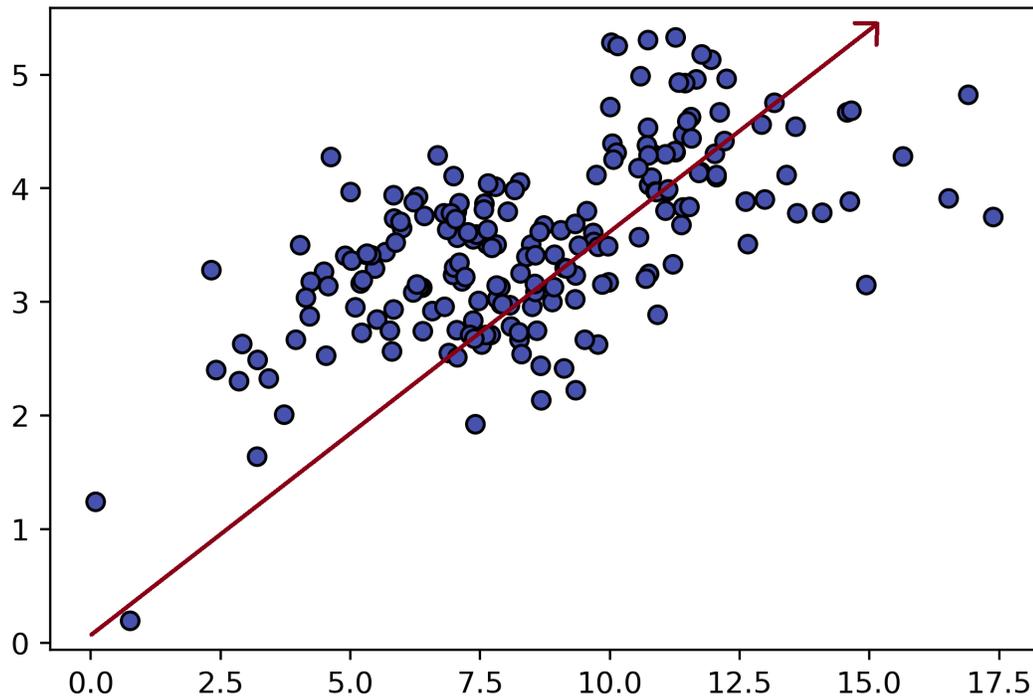


Figura 2.9: Ejemplo de dirección de mayor variación en un conjunto de datos.

ambas variables se incrementan proporcionalmente, por el contrario, si se obtienen valores negativos, quiere decir que a medida que una variable incrementa, la otra decrementa. Si se obtiene un valor de 0, esto indica que son variables independientes.

Para un conjunto de datos de dimensión n , se calcula una matriz de covarianza que tiene n filas y n columnas. Por ejemplo, si tenemos una matriz de 3 dimensiones, la matriz de covarianza resultante es la siguiente:

$$C = \begin{pmatrix} cov(X, X) & cov(X, Y) & cov(X, Z) \\ cov(Y, X) & cov(Y, Y) & cov(Y, Z) \\ cov(Z, X) & cov(Z, Y) & cov(Z, Z) \end{pmatrix}$$

Esta matriz es cuadrada y simétrica sobre la diagonal principal, en su diagonal se encuentran las varianzas en cada dimensión. Sobre esta matriz de covarianzas, se calculan los valores propios (eigenvalues) y vectores propios (eigenvectors). En donde se busca si hay vectores \mathbf{x} diferentes de cero en R^n tales que $A\mathbf{x}$ sea un múltiplo escalar de \mathbf{x} ($A\mathbf{x} = \lambda\mathbf{x}$). El escalar denotado por λ , se llama eigenvalor de la matriz A y el vector \mathbf{x} diferente de

cero se llama eigenvector de A correspondiente a λ , estos eigenvectores son ortogonales, lo que permite representar datos en términos de estos vectores propios. Una matriz puede tener más de un eigenvalor [59].

Este proceso aplicado a imágenes descrito en [60] y en [61], se describe a continuación. El primer paso es convertir cada imagen de dimensión $N \times N$ en un vector de dimensión $N^2 \times 1$ como se muestra en la figura 2.10. Un vector Γ de dimensión $N^2 \times 1$ se busca representar en un espacio dimensional de menor dimensión.

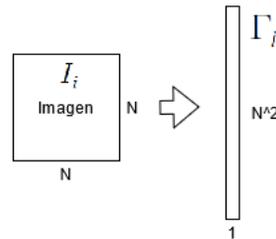


Figura 2.10: Imagen de $N \times N$ representada en forma de vector columna.

Suponiendo que tenemos m imágenes, tendremos una matriz resultante de m columnas y N^2 filas (Fig.2.11). Para poder obtener buenos resultados, las imágenes deben de estar centradas y deben ser de la misma dimensión. Se calcula el promedio (Ecu. 2.10) de cada fila para obtener una imagen promedio Ψ .

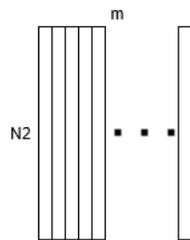


Figura 2.11: Matriz de vectores columna.

$$\Psi = \frac{1}{m} \sum_{i=1}^m \Gamma_i \tag{2.10}$$

A cada imagen Γ_i le sustraemos el promedio Ψ :

$$\Phi = \Gamma_i - \Psi \tag{2.11}$$

Calculamos la matriz de covarianza C en Φ :

$$C = \frac{1}{m} \sum_{n=1}^m \Phi_n \Phi_n^T = AA^T; \quad A = [\Phi_1 \Phi_2 \dots \Phi_m] \quad (2.12)$$

Esto da como resultado una matriz de dimensión $N^2 \times N^2$. Sobre esta matriz se deben calcular los valores y vectores propios, desafortunadamente, es una matriz muy grande que la vuelve impráctica. Si hacemos $A^T A$, obtenemos una matriz de dimensión $m \times m$. Si calculamos los eigenvectores ν_i sobre esta matriz, $A^T A \nu_i = \mu_i \nu_i$. Existe una relación $A^T A \nu_i = \mu_i \nu_i \Rightarrow AA^T A \nu_i = \mu_i A \nu_i \Rightarrow C A \nu_i = \mu_i A \nu_i$ o $C u_i = \mu_i u_i$, con $u_i = A \nu_i$. Por lo tanto, AA^T y $A^T A$ tienen los mismos valores y sus vectores propios se relacionan de la siguiente manera: $u_i = A \nu_i$. Aunque AA^T puede tener hasta N^2 eigenvalores y eigenvectores, y $A^T A$ puede tener hasta M valores y vectores propios, estos últimos son los M más grandes de AA^T .

Una vez calculados los M eigenvalores y eigenvectores, se normaliza u_i ($\|u_i\| = 1$) y mantenemos los k mejores eigenvectores que corresponden a los k eigenvalores más grandes. La reconstrucción se puede realizar de la siguiente manera:

$$\hat{\Phi}_i - \Psi = \sum_{j=1}^k w_j u_j; \quad w_j = u_j^T \Phi_i \quad (2.13)$$

Donde u_j son los eigenfaces y w_j son los pesos de ponderado de cada eigenface para cada imagen. Por tanto, para tener la imagen original debemos de multiplicar cada eigenface por su peso, sumar estas matrices y al resultado se le suma la imagen promedio.

Para que este método se eficiente, las imágenes deben de estar centradas, alineadas, en la misma posición y con expresiones faciales similares.

Si deseamos reconstruir una imagen del conjunto de entrenamiento o proyectar una nueva imagen, convertimos la imagen a vector columna, restamos el promedio a nuestra imagen de entrada, la multiplicamos por la matriz de eigenfaces transpuesta, al resultado lo multiplicamos por la matriz de eigenfaces y por último le sumamos el promedio. Al multiplicar U_i^T por la imagen menos su promedio, obtenemos el vector de ponderado w_j , que nos indica el peso que por el cual se va a multiplicar cada eigenface para poder reconstruir la imagen.

$$\hat{z} = U_i U_i^T (z - \Psi) + \Psi \quad (2.14)$$

2.4. Normalización

La normalización consiste en hacer un escalamiento de los valores de un vector de características, esto se hace para evitar problemas de comparaciones en espacios multi-dimensionales que se pueden presentar debido al rango de valores que pueden tener las características.

Supongamos que tenemos una variable cuyo rango es de 700 a 1900 y otra variable que toma valores entre 0 y 5, lógicamente al graficar estas dos variables los cambios en la variable cuyo valor está entre 0 y 5 serán poco perceptibles. Si se hacen comparaciones de distancia como la distancia euclidiana, la variable de menor rango aportará poca información. Algunos algoritmos como es el caso de k - NN utilizan la distancia euclidiana para hacer comparaciones de similitud entre muestras para clasificación por lo que es recomendable realizar la normalización de las características para que todas ellas varíen en un mismo rango.

La normalización produce valores sin unidades de medida que nos permiten poder comparar elementos de distintas variables y distintas unidades de medida. Si no hay una operación de normalización de datos entre diferentes características, las características más grandes siempre dominarán. Las normalizaciones más comunes son *MinMax* en donde el valor mínimo es 0 y el valor máximo es 1 y *mean0* cuyo promedio es 0 y su desviación estándar es igual a 1. La primera es más sensible a *outlayers*, estos son valores no comunes que se salen del rango en el cual la mayoría de los demás valores de esa variable se encuentran. Por ejemplo, si tenemos una variable que toma valores entre 70 y 90 y hay una muestra que tiene un valor de 5 en esta característica, afectará la normalización porque el valor mínimo de cero solo será tomado por el 5 y los demás valores estarán cercanos al 1, por lo que solo es adecuado para situaciones en las que los valores de los datos están relativamente concentrados. Con la distribución normal no pasa esto, los datos procesados obedecerán la distribución normal estándar, el rango de valores después del procesamiento no es $[0,1]$, esta distribución sirve para filtrar valores atípicos, pero se requiere que la distribución de los datos originales sea aproximadamente gaussiana. Esto no quiere decir que una sea mejor que la otra, sino que dependerá de cómo estén distribuidos los datos.

Las características extraídas de una muestra pueden ser representadas como un vector fila \mathbf{x} de m elementos, donde m es el número de características extraídas. Por lo tanto, una muestra se puede ver como un punto $\mathbf{x} = [x_1, \dots, x_m]$ en el espacio de características de m

dimensiones. Los vectores de características de todas las muestras pueden ser almacenados en una matriz \mathbf{X} de tamaño $N * m$, donde N es el número de muestras. La j -ésima columna de \mathbf{X} , \mathbf{x}_j está conformada por los valores que puede tomar la característica x_i para todas las muestras. El elemento x_{ij} representa a la característica x_i de la muestra j . La normalización se hace mediante

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i} \quad (2.15)$$

para $i = 1, \dots, m$ y $j = 1, \dots, N$, donde μ_i y σ_i son el promedio y la desviación estándar respectivamente de \mathbf{x}_i . Las características normalizadas tienen un promedio de 0 y una desviación estándar de 1 [62].

2.5. Selección de Características

Los vectores de características generalmente son grandes y esto hace que los algoritmos sean más tardados, además algunas de estas características no son relevantes en el proceso de clasificación y por el contrario pueden afectar la discriminación. El proceso de selección de características es altamente complejo ya que se pueden extraer infinidad de características de una imagen, pero es difícil reconocer cuales son realmente necesarias.

Una forma de observar que tan discriminantes son las características, es observando la distribución de frecuencia de cada clase y definiendo un factor umbral de discriminación para que se consideren únicamente las características que superen este umbral. Se pueden obtener tres escenarios diferentes, características malas, buenas y excelentes (Fig. 2.12). En el primer caso, es prácticamente imposible realizar una correcta clasificación. En el segundo caso se pueden obtener buenos resultados utilizando un clasificador. En el último caso no es necesario un clasificador y basta con definir un umbral para discriminar entre clases.

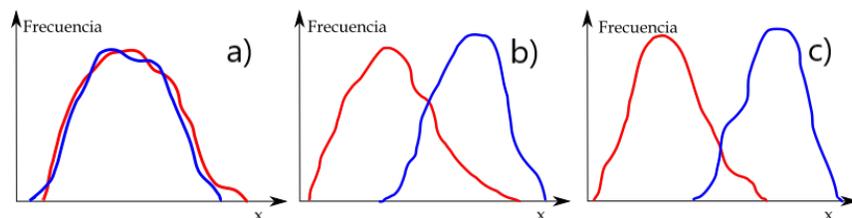


Figura 2.12: a) Mala separación entre clases. b) Separación entre clases buena. c) Excelente separación entre clases.

Supongamos que tenemos dos clases ω_0 y ω_1 y deseamos saber que tan buena es una

característica para discriminar entre estas clases. Entonces, para cada característica x , obtenemos sus funciones de distribución y las graficamos. Para valores reales, generalmente se obtendrá funciones que se superponen entre ellas. En la figura 2.13 se puede observar estas funciones de manera separada (Fig.2.13a, Fig.2.13b) y también en un mismo gráfico (Fig.2.13c), claramente esta característica no es una característica que pueda ser utilizada para clasificación.

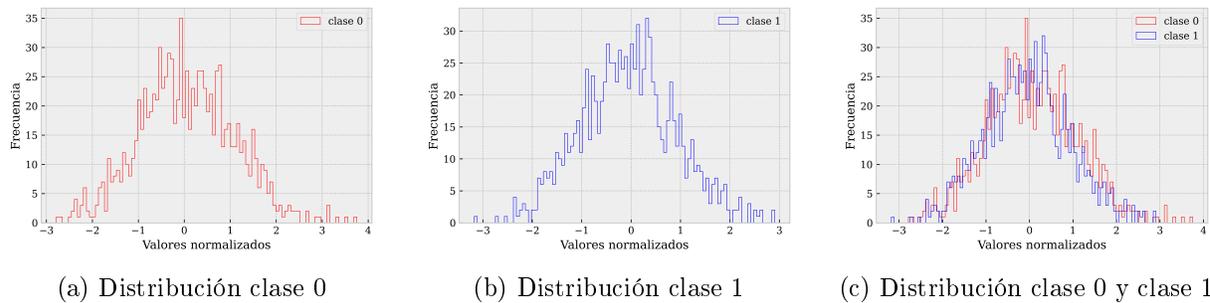


Figura 2.13: Funciones de distribución para una característica x .

Idealmente para una correcta discriminación las funciones de distribución deben de estar separadas, como se mencionó anteriormente, esto no es posible y en la realidad las características que pueden ser utilizadas para discriminación se asemejan a la figura 2.14. Aunque ocurre un traslape, utilizando varias características con este comportamiento, se pueden obtener buenos resultados, aunque no se reconocerán ambas clases perfectamente. En la figura 2.15 se ejemplifica la clasificación entre dos clases utilizando dos características.

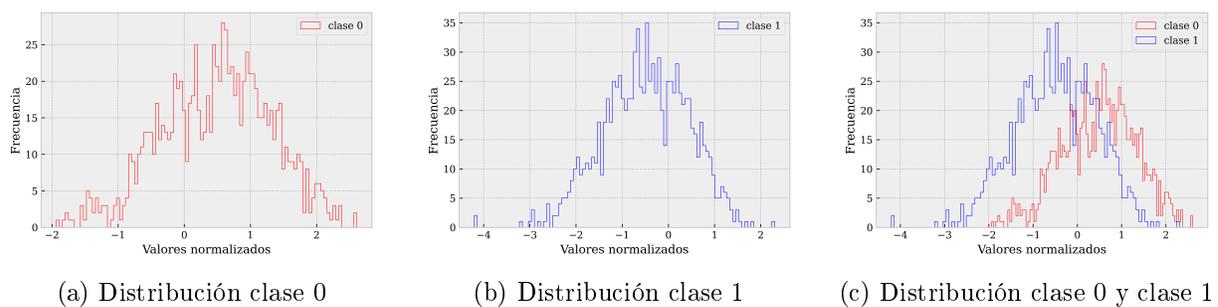


Figura 2.14: Funciones de distribución para una característica x real.

Otro problema que surge en la selección de características es cuantas de ellas seleccionar. Para evitar *the curse of the dimensionality* se recomienda utilizar pocas características. Al incrementarse el número de características, el volumen del espacio de características aumenta exponencialmente. También si se tienen más características que observaciones el

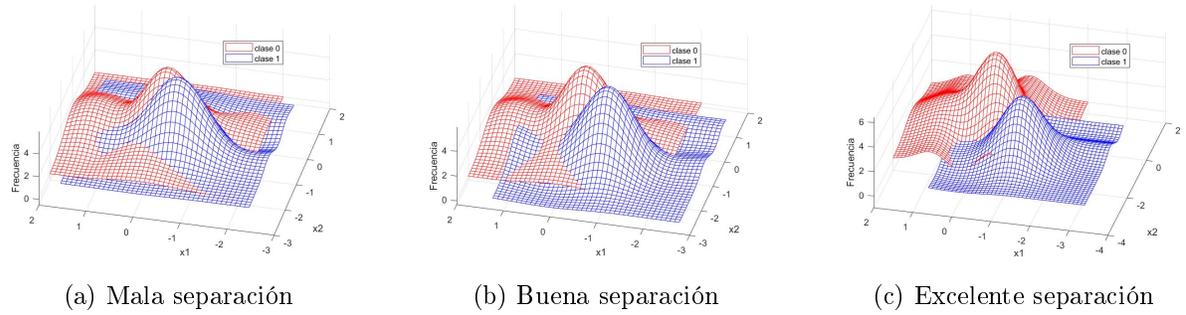


Figura 2.15: Distribución de 3 pares de características (x_1, x_2) .

modelo se puede sobre ajustar.

La idea principal en la selección de características es seleccionar un subconjunto de p características ($p \leq m$) que produzca el menor error de clasificación. Las p características seleccionadas se agregan a un nuevo vector de p elementos $\mathbf{z} = [z_1 \dots z_p]$. Los vectores de todas las muestras pueden ser almacenados en una nueva matriz \mathbf{Z} de dimensión $N * p$. Este proceso se ejemplifica en la figura 2.16 para $m = 14$ y $p = 4$. Las p características seleccionadas son columnas s_1, s_2, \dots, s_p de \mathbf{X} como se puede apreciar en la figura con las columnas indicadas con un color diferente. El vector de etiquetas no se modifica, solo se hace la selección de características.

El proceso de selección de características se puede establecer de la siguiente manera, dado el conjunto de características de N muestras (\mathbf{X}), y las etiquetas de cada muestra (\mathbf{y}) se busca un conjunto de características $\mathbf{s} = (s_1, s_2, \dots, s_p)$ que maximice la separación entre clases $\mathbf{J}(\mathbf{X}, \mathbf{s}, \mathbf{y})$, que es un problema de optimización [62]

$$\hat{\mathbf{s}} = \operatorname{argmax} J(\mathbf{X}, \mathbf{s}, \mathbf{y}); \quad \mathbf{s} \subset Q \tag{2.16}$$

donde $Q = 1, 2, \dots, m$ es el conjunto de índices que puede tomar \mathbf{s} .

Hay distintas aproximaciones que pueden ser usadas para medir la separabilidad. Una de ellas es utilizando el criterio de Fisher que busca la menor variación dentro de la clase y la mayor separación entre clases. Inicialmente se utilizó Fisherfaces para realizar la separación, el problema es que, con este método, hay como máximo $c - 1$ valores propios generalizados distintos de cero, por lo que un límite superior en m es $c - 1$, donde c es el número de clases [63], para el algoritmo de clasificación propuesto basado en kNN no es benéfico porque al ser solo dos clases únicamente queda una característica, esto le

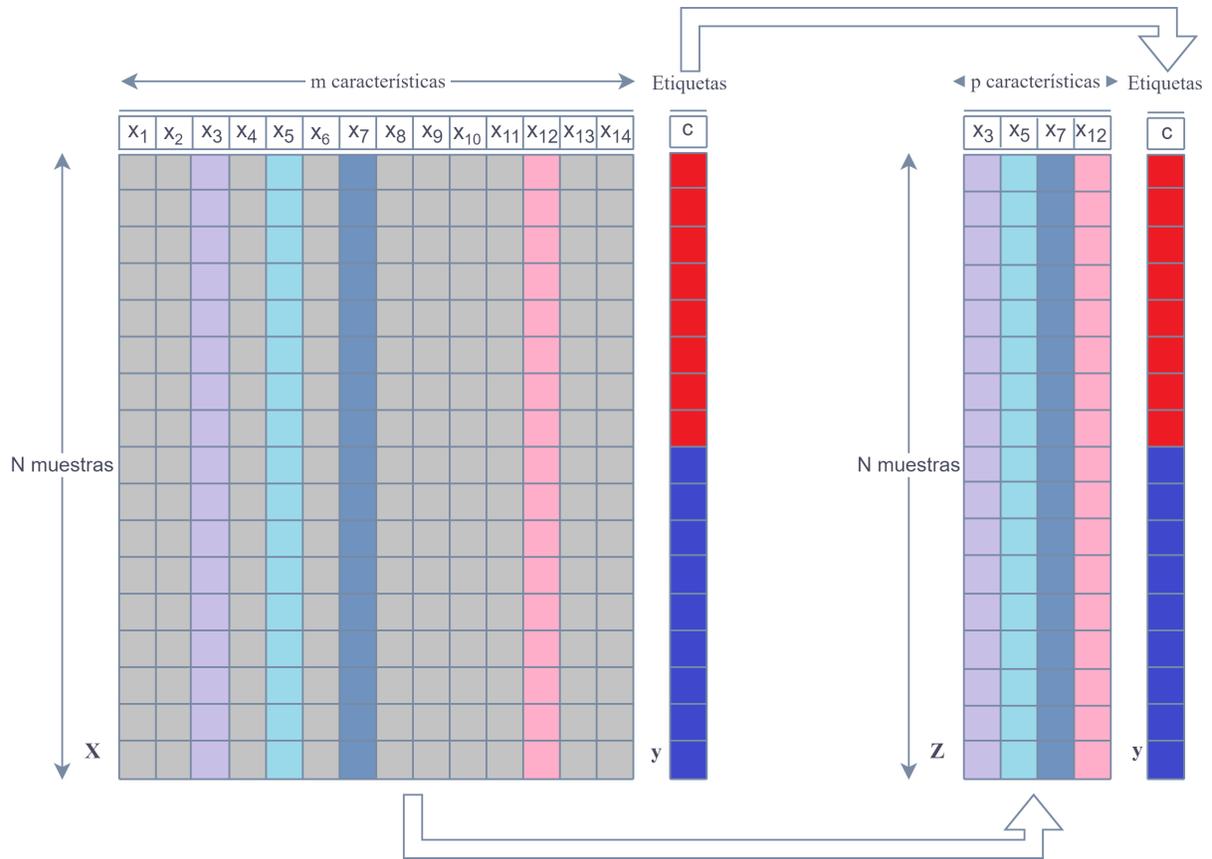


Figura 2.16: Selección de características. De las m características disponibles, se seleccionan p .

quita libertad de movimiento al algoritmo. En este trabajo, basándonos en Fisherfaces la selección de características se calcula un factor de discriminación estadístico. Utiliza el promedio en de cada clase $\mu_{i_{c0}}, \mu_{i_{c1}}$ para cada característica x_j . De esta manera obtenemos los “centros” de las distribuciones

$$\mu_i = \frac{1}{N} \sum_{j=1}^N x_{ij} \quad (2.17)$$

donde i es la característica y j es el valor de esta característica para cada muestra. También, es necesario determinar el ancho de la campana (Fig.2.17) mediante al cálculo de la varianza que nos indica que tan dispersos están los datos

$$\sigma_i^2 = \frac{1}{N} \sum_{j=1}^N (x_{ij} - \mu_i)^2 \quad (2.18)$$

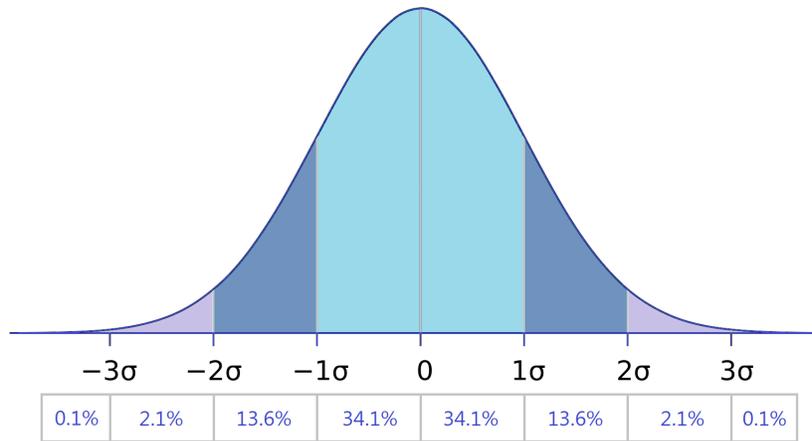


Figura 2.17: Visualización de la desviación estándar para una distribución normal.

El factor de discriminación J se calcula con la formula 2.19. Al restar los promedios obtenemos la separación entre clases. Elevamos al cuadrado el resultado de la resta para eliminar el signo y el resultado lo dividimos entre la suma de las varianzas

$$J_i = \frac{(\mu_{i_{c0}} - \mu_{i_{c1}})^2}{\sigma_{i_{c0}}^2 + \sigma_{i_{c1}}^2} \quad (2.19)$$

donde $\mu_{i_{c0}}$ es el promedio de la característica i de la clase 0, $\mu_{i_{c1}}$ es el promedio de la característica i de la clase 1, $\sigma_{i_{c0}}^2$ es la varianza de la característica i de la clase 0 y $\sigma_{i_{c1}}^2$ es la varianza de la característica i de la clase 1.

Entre mayor sea la separación entre clases y menor la dispersión, mayor será el factor J . El siguiente paso es definir un umbral Trd para seleccionar las características más relevantes $j_i > Trd$. De esta manera se obtienen las características que separan mejor el conjunto de datos.

Capítulo 3

Desarrollo del Sistema Automático

Este capítulo describe los procesos realizados para detectar neumonía en imágenes de rayos-X. Se parte de un banco de imágenes a las que se les aplica una mejora de contraste. El siguiente paso consiste en crear un eigenespacio con las imágenes de entrenamiento, la proyección de cada imagen sobre el eigenespacio produce vectores de peso de ponderación de las eigenfaces que son utilizados como vectores de características. Las características son normalizadas para posteriormente seleccionar las características más relevantes para clasificación. Por último, estas características seleccionadas son utilizadas para entrenar el algoritmo propuesto basado en k-NN para realizar la clasificación. La complejidad del algoritmo radica únicamente en el entrenamiento, ya que para la clasificación se aplica k-NN tradicional ponderado tomando como prototipos los prototipos obtenidos durante la etapa de entrenamiento y que están ubicados de tal manera que generalizan el conjunto de entrenamiento.

3.1. Estructura General

El diagrama de la figura 3.1 muestra de manera simplificada la estructura general del sistema completo para la etapa de entrenamiento y la etapa de prueba. Las imágenes utilizadas son la base de datos COVID-19 radiography database [36]. Se utilizaron estas imágenes debido a que las clases están balanceadas y son combinaciones de otras bases de datos, de esta manera, se obtienen imágenes variadas.

La mejora de contraste se realizó mediante la ecualización del histograma que se describe en el capítulo 2. El recorte no es inteligente, no toma en cuenta el área real que ocupan los pulmones dentro de la imagen, para realizar este recorte, de manera visual se observó que porción de los extremos de la imagen era posible eliminar de manera que en la

“mayoría” de las imágenes se recortara el área de los pulmones. La redimensión se definió mediante prueba y error de manera que el costo computacional fuera el menor posible sin afectar la clasificación por pérdida de información.

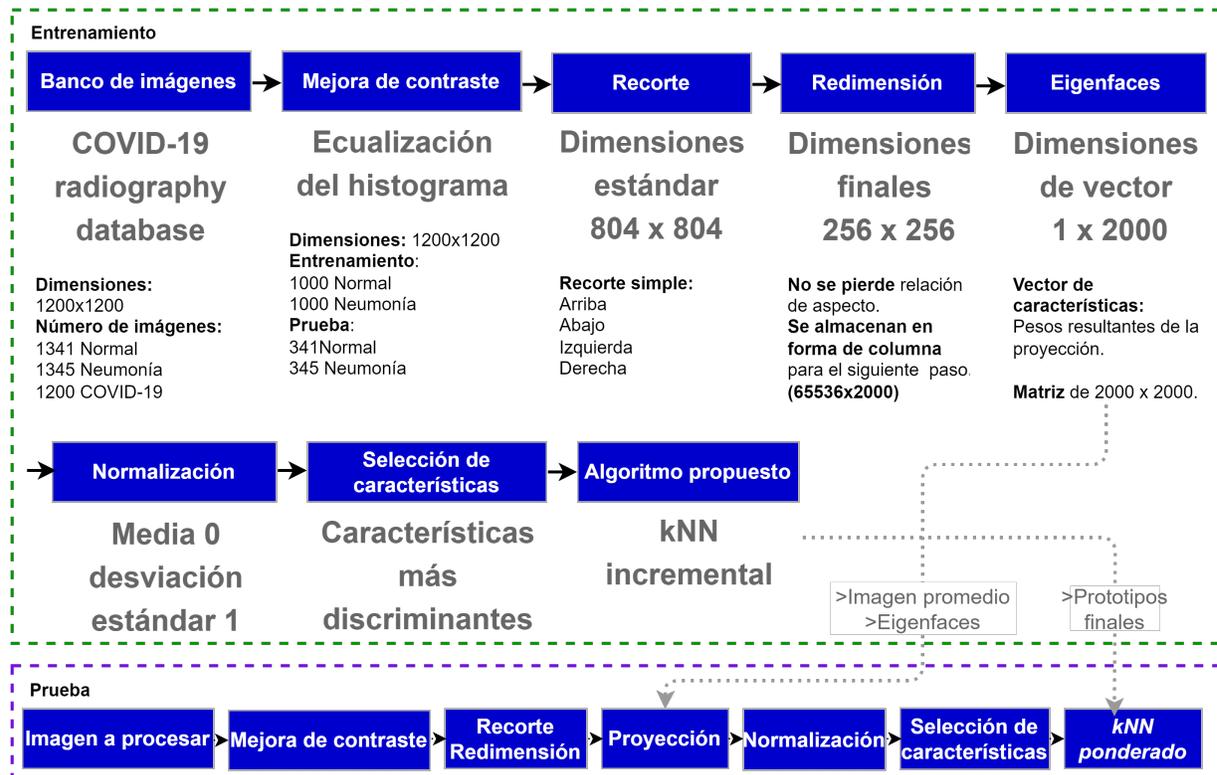


Figura 3.1: Diagrama general del sistema (entrenamiento y prueba).

Para la extracción de características se probaron varias técnicas: características Haralick, Bolsas de Palabras, Fisherfaces y Eigenfaces, siendo esta última la que produjo mejores resultados (ver apéndice B). Para aplicar el método de Eigenfaces es necesario colocar las imágenes en forma de vector columna y con ellas formar una matriz, la matriz resultante de este proceso tiene una dimensión de 65536x2000, donde las filas son los píxeles de las 2000 imágenes de entrenamiento. Sobre esta matriz se aplica eigenfaces y los pesos resultantes de la proyección de cada una de las imágenes de entrenamiento, se utilizan como vectores de características, estos vectores tienen una dimensión de 2000. El siguiente paso es la normalización que consiste en escalar los valores de cada característica de manera que tengan una media de cero y una desviación estándar de 1.

Posteriormente se seleccionan las características que mejor separen las dos clases, para esto se ideó un método basado en el discriminante de Fisher de manera que se obtienen

alrededor de 50 características. Todos estos algoritmos se describen con más detalle en la sección de marco teórico. Una vez obtenidas estas características, el último paso es el entrenamiento para la clasificación que se realiza mediante un nuevo algoritmo que se describe en la siguiente sección. Cabe mencionar que la complejidad del algoritmo radica únicamente en la etapa de entrenamiento que obtiene la posición de los prototipos que mejor generalizan al conjunto de entrenamiento, para la etapa de prueba, se aplica k-NN tradicional ponderado, de manera que se obtiene un resultado muy rápido.

La interfaz gráfica aplica los mismos procesos en la imagen que la etapa de prueba de la figura 3.1, la única diferencia es que la interfaz permite hacer un recorte cuadrado manual con el ratón y el teclado de la computadora de manera que se seleccione lo más cercano posible el área del pulmón sin perder la relación de aspecto.

Los resultados mostrados en la interfaz gráfica pueden ser: estado normal o presencia de neumonía. Para arrojar estos resultados, el sistema aplica k-NN ponderado tradicional y si la etiqueta predicha es normal, se despliega este resultado en la ventana de interfaz. Por el contrario, si la etiqueta estimada es presencia de neumonía se realiza un cálculo de probabilidad de neumonía tomando en cuenta la votación que arrojó cada clase:

$$P = \frac{v1}{v0 + v1} * 100 \quad (3.1)$$

donde $v1$ representa la votación de la clase neumonía y $v0$ es el valor de la votación de la clase normal. De esta manera, se obtendrán valores de probabilidad mayores a 50 que dependen de los vecinos que rodean a la muestra a clasificar.

3.2. k-NN Propuesto

Como se mencionó en capítulos anteriores, los problemas que surgen en el algoritmo k-NN al aumentar el número de muestras son que los requisitos de almacenamiento y el costo computacional también aumentan debido a que se hace una comparación de similitud por proximidad entre la muestra a clasificar y cada una de las muestras de referencia disponibles en el conjunto de datos. En este sentido, se han presentado diversas soluciones para enfrentar este problema mediante la selección de prototipos representativos del conjunto de muestras disponibles (selección de prototipos) y mediante la creación de nuevos prototipos aplicando múltiples operaciones sobre el conjunto de datos original (generación de prototipos) [64].

Las mejoras que se han planteado se enfocan en aspectos como: la reducción de instancias, el mejoramiento de la exactitud en la clasificación y en la aceleración del proceso. En este trabajo se busca mejorar la exactitud y mediante la generación de un conjunto reducido prototipos acelerar la clasificación. Además, para conjuntos de datos grandes, repetir el proceso de entrenamiento no es viable por lo que se busca que conforme arriben nuevas muestras no se repita el entrenamiento completo, sino que solo se introduzcan las nuevas muestras al algoritmo. Este tipo de aprendizaje se conoce como aprendizaje incremental.

En este trabajo presentamos un nuevo algoritmo de clasificación basado en k-NN, que toma muestras iniciales de manera aleatoria como prototipos que cambian durante una etapa de entrenamiento con nuevas muestras que arriban de forma secuencial. Cada prototipo en el algoritmo además de su vector de características, y su respectiva etiqueta de clase, tiene asociado un registro o memoria a corto plazo en el cual se irán almacenado copias de las muestras para las cuales dicho prototipo resultó ser el más cercano. Dicha memoria es de tipo FIFO y tiene un tamaño limitado. Además, cada prototipo posee otra memoria más, también de tipo FIFO, la cual almacenará valores de una variable que nosotros hemos denominado CLS (Classification Strength) y que representa en un rango de 0 a 1 la intensidad de la clasificación de la muestra otorgada por el algoritmo k-NN:

$$CLS = 1 - \frac{Vl}{Vw} \quad (3.2)$$

donde Vl es la votación para la clase perdedora, y Vw la votación para la clase ganadora.

El primer paso del algoritmo es aplicar k-NN a cada una de las muestras del conjunto de entrenamiento para obtener la fuerza de clasificación, tomando como prototipos todas las muestras restantes y si este valor es mayor a 0.7 (para el caso de las radiografías), al terminar de aplicar k-NN a todo el conjunto de entrenamiento, estas muestras se eliminan. Este proceso de podado se realiza para hacer menos pesado el entrenamiento tomando únicamente muestras de la frontera (algoritmo 2).

En el siguiente paso, el algoritmo se inicializa con m muestras válidas por clase del conjunto podado que son tomadas como prototipos iniciales. Visualizando el funcionamiento del entrenamiento en el espacio de características F-dimensional (con F siendo el número de características) donde yacen las muestras, cada vez que una nueva muestra se presenta, mientras ésta no sea bien clasificada en el sentido de k-NN (por los prototipos existentes

Algoritmo 2: Podado

```

Data:  $\mathcal{T}$ ,  $K$ ,  $thrPr$ 
 $\mathcal{TP} \leftarrow \emptyset$ ;
for  $\langle s_t, l_t \rangle \in \mathcal{T}$  do
   $CLS = classification\_strength(s_t, \mathcal{T}, K)$ ;
  if  $CLS \leq thrPr$  then
     $i \leftarrow i + 1$ ;
     $\mathcal{TP}_i \leftarrow \langle s_t, l_t \rangle$ ;
  end
end
return ( $\mathcal{TP}$ )

```

en ese momento), el prototipo que resulte más cercano se acercará en pasos pequeños a la muestra para intentar clasificarla correctamente, siempre y cuando ninguna de las muestras en las memorias resulte mal clasificada debido a dicho acercamiento. Este proceso se realizará con los k vecinos más cercanos a la muestra. En caso de que los acercamientos provoquen que la muestra sea bien clasificada, una copia de dicha muestra será agregada a la memoria del prototipo más cercano. Luego se verificará que las muestras almacenadas en las memorias de los prototipos cumplan con el requisito de pertenecer a la memoria de su prototipo más cercano. En caso de no cumplirse dicho requerimiento, se realizará una redistribución apropiada de las muestras almacenadas. Finalmente, se continuará con la siguiente muestra de entrenamiento.

Si los pasos de acercamiento de un prototipo provocaran una incorrecta clasificación de alguna de las muestras almacenadas en las memorias de los prototipos, antes de alcanzar la clasificación correcta de la muestra de entrenamiento, entonces el prototipo “acercado” dará un paso atrás regresando a su posición inmediata anterior, y será el siguiente vecino más cercano (de la misma clase) quien continuará dando pasos de acercamiento. Este proceso continuará hasta el último vecino más cercano k , mientras no se haya logrado la clasificación correcta de la muestra de entrenamiento. En caso de que el acercamiento del último vecino más cercano provoque una clasificación incorrecta de alguna de las muestras en las memorias, todos los k vecinos acercados (incluyendo el último), regresarán a sus posiciones iniciales.

Por el contrario, si al presentarse una nueva muestra de entrenamiento, ésta es bien clasificada (usando k -NN), entonces la posición del prototipo más cercano se actualizará como el promedio de las posiciones de las muestras dentro de su memoria. También, se almacenará el valor CLS de dicha clasificación en la memoria de valores CLS del prototipo.

Las muestras lejos de la frontera presentan un CLS mayor en comparación con las muestras en la frontera. La explicación de esto es que los prototipos en la zona de la frontera de clasificación clasifican con menor fuerza a las muestras de su vecindad. La figura 3.2 ilustra el caso cuando que fue incorrectamente clasificada desde el principio, es alcanzada por sus prototipos vecinos para intentar una clasificación correcta de ésta. En el primer paso del segundo prototipo, la muestra ha resultado bien clasificada debido al acercamiento de los prototipos (siempre sin provocar algún error de clasificación en las muestras almacenadas en las memorias de éstos). Por otra parte, la figura 3.3 ilustra el caso en que antes de lograr una correcta clasificación de la muestra, el movimiento de los prototipos ha ocasionado una incorrecta clasificación de una muestra dentro de alguna de las memorias de los prototipos.

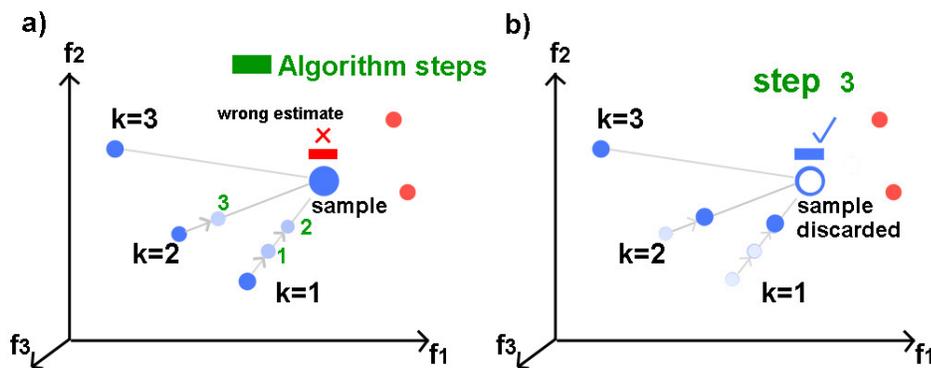


Figura 3.2: a) Una muestra que fue incorrectamente clasificada desde el principio. b) Se alcanza una correcta clasificación con el acercamiento de sus prototipos vecinos.

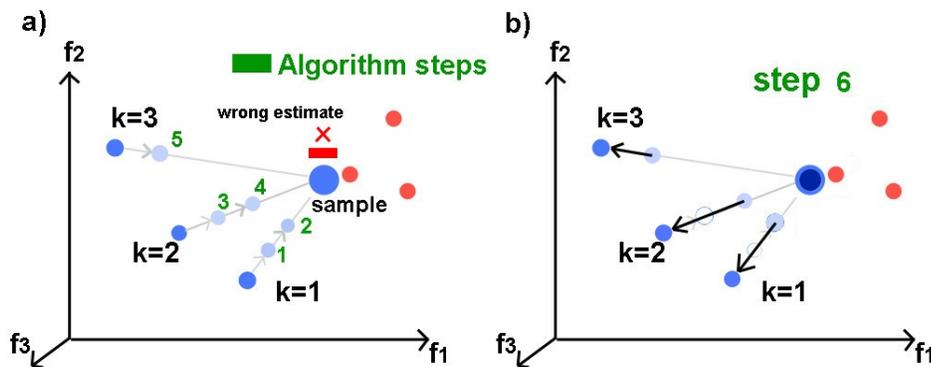


Figura 3.3: a) Una nueva muestra, que fue incorrectamente clasificada desde el principio, es alcanzada por sus prototipos vecinos para intentar una clasificación correcta de ésta. b) La muestra sigue resultando equivocadamente clasificada a pesar del acercamiento de los prototipos. Sin embargo, el proceso de acercamiento es interrumpido cuando una muestra almacenada en alguna memoria resulta incorrectamente clasificada debido al movimiento del tercer prototipo. En este caso, los prototipos acercados son regresados a sus posiciones iniciales.

3.2.1. Algoritmo k-NN propuesto

Denotamos al conjunto de entrenamiento como $\mathcal{T} = \langle \mathbf{s}_1, l_1 \rangle, \langle \mathbf{s}_2, l_2 \rangle, \dots, \langle \mathbf{s}_T, l_T \rangle$, donde $\langle \mathbf{s}_t, l_t \rangle$ es una muestra en particular compuesta por un vector de características \mathbf{s}_t , siendo t una muestra en el rango de 1 a T , con su respectiva etiqueta l_t . El algoritmo se inicializa con m prototipos definidos por el usuario de cada clase (algoritmo 3). El conjunto de muestras de entrenamiento podado se define como \mathcal{TP} .

Algoritmo 3: Prototype initialization

Data: \mathcal{TP} , m
 $P \leftarrow \emptyset$; $c1 \leftarrow 0$; $c2 \leftarrow 0$;
for $\mathbf{s}_i \in \mathcal{TP}$ **do**
 if $l_i = 1$ & $c1 \leq m$ **then**
 $c1 \leftarrow c1 + 1$;
 $P_i \leftarrow \langle \mathbf{s}_i, l_i \rangle$;
 end
 if $l_i = 2$ & $c2 \leq m$ **then**
 $c2 \leftarrow c2 + 1$;
 $P_i \leftarrow \langle \mathbf{s}_i, l_i \rangle$;
 end
 if $c1 = m$ & $c2 = m$ **then**
 return (P)
 end
end
return (P)

Representamos cada prototipo como una 4-tupla: $\langle p_n, M_n, CLSM_n, l_n \rangle$, donde p_n es la posición en el espacio F -dimensional del prototipo, p. ej. $p_n = (f_1, f_2, \dots, f_F)$, donde F es el número de características. $CLSM_n$ es una memoria en donde se almacena la fuerza de clasificación de las últimas muestras almacenadas. l_n es su respectiva etiqueta de clase. M_n representa una memoria *FIFO* para almacenar hasta J réplicas de las muestras, donde cada replica muestra se representa como $\langle \mathbf{s}_j, l_j \rangle$ siendo $\mathbf{s}_j = (f_1, f_2, \dots, f_F)$ el vector de características y l_j su etiqueta, con $j = 1, 2, \dots, J$,

$$M_n = \langle \langle \mathbf{s}_1, l_1 \rangle, \langle \mathbf{s}_2, l_2 \rangle, \dots, \langle \mathbf{s}_J, l_J \rangle \rangle. \quad (3.3)$$

Cuando una nueva muestra $\langle \mathbf{s}_q, l_q \rangle$ se añade a M_n , esta se almacenará en la primera posición, y el resto de las muestras en la memoria se desplazará a la derecha eliminando

de esta manera a la muestra más antigua,

$$M_n = \langle (s_q, l_q), (s_1, l_1), (s_2, l_2), \dots, (s_{J-1}, l_{J-1}) \rangle. \quad (3.4)$$

Algoritmo 4: Main

Data: \mathcal{TP} , K , J , $JCLS$, P

for $s_i \in \mathcal{TP}$ **do**

$EL, n1, n2, CLS \leftarrow \text{k-NN}(s_i, l_i, K, P)$;

if $l_i = 1$ **then**

$n \leftarrow n1$;

else

$n \leftarrow n2$;

end

$k1 \leftarrow n[1]$;

if $EL = l_i$ **then**

$\text{store_sample_in_prototype}(s_i, l_i, CLS, P_{k1})$;

$p_backup \leftarrow P_{k1}$;

$mle_old \leftarrow \text{MLE}(P, K)$;

$P_{k1} \leftarrow \frac{1}{\text{len}(M_{k1})} \sum_{j=1}^{\text{len}(M_{k1})} s_j, \quad s_j \in M_{k1}$;

$mle \leftarrow \text{MLE}(P, K)$;

if $mle_old < mle$ **then**

$P_{k1} \leftarrow p_backup$

end

if $\text{len}(M_{k1}) > J$ **then**

$M_{k1} \leftarrow \langle (s_i, l_i), (s_1, l_1), (s_2, l_2), \dots, (s_{J-1}, l_{J-1}) \rangle$;

end

else

$\text{approaching_prototypes}(P, n, s_i, l_i, J, K)$;

end

end

return (P)

Algoritmo 5: Aproximar prototipos

Data: $P, n, \mathbf{s}_i, l_i, J, K$
 $CLS0 \leftarrow 1; k \leftarrow 1; approaching \leftarrow True; permitted_steps \leftarrow 21; division \leftarrow 30; steps \leftarrow 0;$
 $k_original_prototypes \leftarrow backup_prototypes(P, K);$
while $approaching = True$ **do**
 $nk \leftarrow n[k];$
 $pback \leftarrow P_{nk};$
 if $first_step$ **then**
 $delta = (muestra - prototipo) / division$
 end
 $mle_old \leftarrow MLE(P, K);$
 $P_{nk} = P_{nk} + delta;$
 $mle \leftarrow MLE(P, K);$
 $estimated_label, CLS \leftarrow kNN(\mathbf{s}_i, l_i, K, P);$
 if $(estimated_label = l_i) \ \& \ (mle \leq mle_old)$ **then**
 $store_sample_in_ptototype(\mathbf{s}_i, l_i, CLS, P_{nk});$
 $mem_samples_redistribution();$
 $approaching \leftarrow False;$
 else if $(estimated_label \neq l_i) \ \& \ (mle \leq mle_old)$ **then**
 $steps \leftarrow steps + 1;$
 if $(CLS < CLS0) \ \& \ steps < permitted_steps$ **then**
 $CLS0 \leftarrow CLS;$
 $approaching \leftarrow True;$
 else
 $k \leftarrow k + 1; steps \leftarrow 0; approaching \leftarrow True;$
 if $k > K$ **then**
 $approaching \leftarrow False;$
 $restore_prototypes(k_original_prototypes);$
 end
 end
 else if $mle > mle_old$ **then**
 $P_{nk} \leftarrow pback; k \leftarrow k + 1; pasos \leftarrow 0;$
 if $k > K$ **then**
 $approaching \leftarrow False;$
 $restore_prototypes(k_original_prototypes);$
 end
end

En algunos pasos del algoritmo es necesario revisar si las réplicas almacenadas en la memoria del prototipo continúan siendo clasificadas correctamente a medida que los prototipos evolucionan. Para hacer esto, la etiqueta de cada replica en memoria se estima para cada prototipo, esta estimación se realiza con el mismo k-NN, usando los prototipos creados. Cuando una o más muestras son incorrectamente estimadas, el proceso retornará el número de muestras incorrectas la función se define como $MLE()$.

Cuando una nueva muestra de entrenamiento \mathbf{s}_f que entra al algoritmo es mal clasificada, se realiza un proceso de aproximación gradual de los prototipos p_k (donde $k = 1, 2, \dots, K$) más cercanos a la muestra, hasta que la muestra sea estimada correctamente. En el caso de que esto no suceda, la muestra se agrega como un nuevo prototipo, esta función se llama *approaching_prototypes()* (algoritmo 5).

El proceso de acercamiento es paso a paso y en cada paso se verifica si hay una mejora en el parámetro CLS y si la muestra es correctamente clasificada. Si ocurre una estimación correcta de la etiqueta, las nuevas posiciones de los prototipos acercados serán definitiva y una réplica de la muestra se almacenará en el prototipo más cercano. Si es necesario, se llevará a cabo una reasignación de las réplicas en las memorias de todos los prototipos. Denotamos este proceso como *mem_samples_redistribution()*. El algoritmo principal se muestra en el algoritmo 4.

3.3. Interfaz Gráfica

Para facilitar la detección de neumonía se realizó una interfaz gráfica en Python utilizando la biblioteca de tkinter. En la imagen 3.4 se muestra la ventana principal que aparece al ejecutar el programa. En el menú archivo se tienen las opciones de **Abrir** y **Guardar**. El primero nos permite cargar una imagen utilizando el explorador de archivos (Fig.3.7a) y el segundo guardar la imagen en una ubicación seleccionada en el explorador de archivos (Fig.3.7b).

Es posible cerrar la aplicación desde el menú archivo o pulsando el botón de salir de la interfaz al igual que al cerrar la ventana. Se buscó que la interfaz fuera intuitiva y fácil de usar por lo que además de los menús se cuenta con un botón para procesar y una pantalla que muestra el resultado, esta pantalla cambia de color dependiendo del resultado. Del lado izquierdo se muestra la imagen cargada y se debe recortar para seleccionar el área de los pulmones. También se muestra un mensaje de bienvenida, un mensaje que indica como

cargar una imagen en la interfaz, un mensaje que refiere consultar la ayuda y un último mensaje que menciona que la tasa de reconocimiento es de alrededor de 92 %.

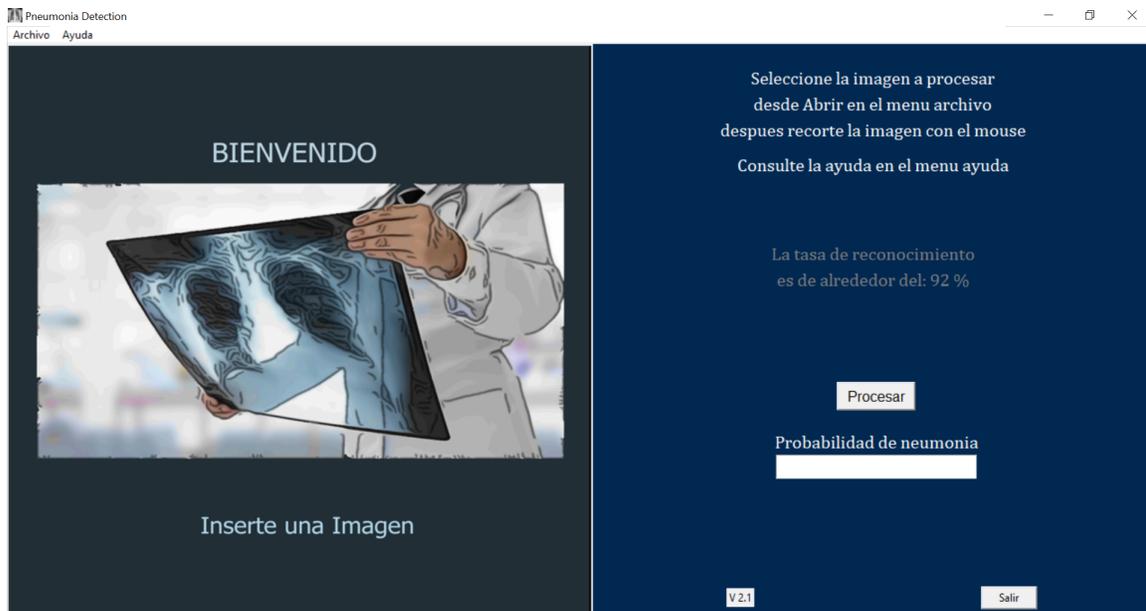


Figura 3.4: Ventana principal Interfaz Gráfica.

La interfaz cuenta con un menú de archivo y aun menú de ayuda como se muestra en las imágenes 3.5 y3.6. Estos se integraron a la interfaz para tener un aspecto más simple y ordenado.

El menú ayuda, despliega la **Ayuda** para procesar una imagen (Fig.3.8).La ayuda contiene el siguiente mensaje: *“Para iniciar abra el menú archivo ->abrir, a continuación, se abrirá una ventana del explorador en la que se puede seleccionar una imagen en formato .jpeg, .jpg o .png. Primero seleccione la extensión de la imagen, de otra manera no se mostrarán imágenes en el explorador de archivos. Después de seleccionar la extensión y la imagen deseada, dar clic en abrir y la imagen seleccionada se mostrará en la interfaz. Si la imagen que se muestra es la que usted desea diagnosticar, el siguiente paso es recortar la zona de los pulmones. Para esto, presione el botón izquierdo del ratón y manténgalo presionado para seleccionar la zona deseada. Si desea volver a realizar la selección, oprima el botón izquierdo dos veces y la selección actual se borrará permitiendo hacer una nueva selección. También pude ajustar la posición del cuadrado con las teclas izquierda, derecha, arriba y abajo del teclado. Una vez seleccionada la imagen de clic en el botón de procesar y al finalizar el análisis, se mostrará la probabilidad de neumonía con el fondo en color rojo o si está en un estado normal con el fondo en color verde.”*

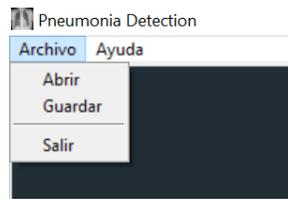


Figura 3.5: Menú Archivo.

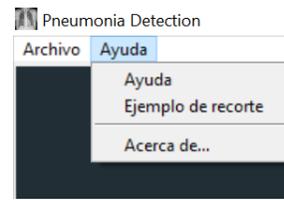
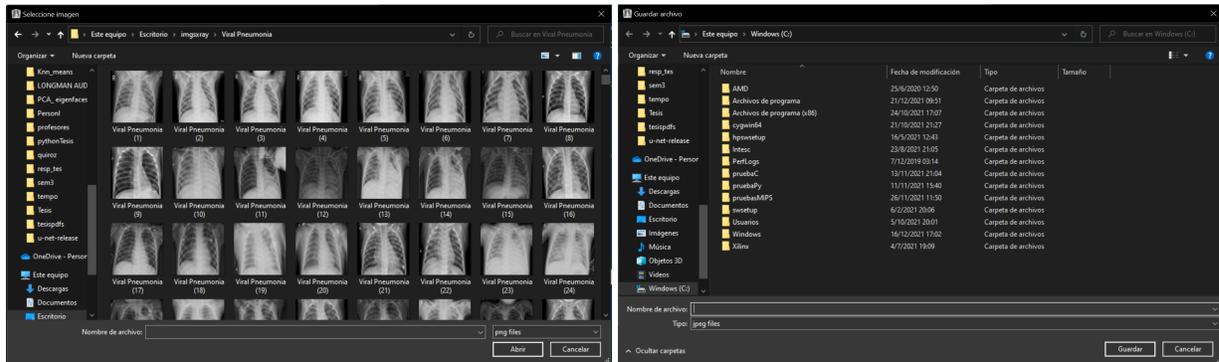


Figura 3.6: Menú Ayuda.



(a) Abrir imagen

(b) Guardar imagen

Figura 3.7: Explorador de archivo que se abre al seleccionar abrir o guardar en el menú archivo.

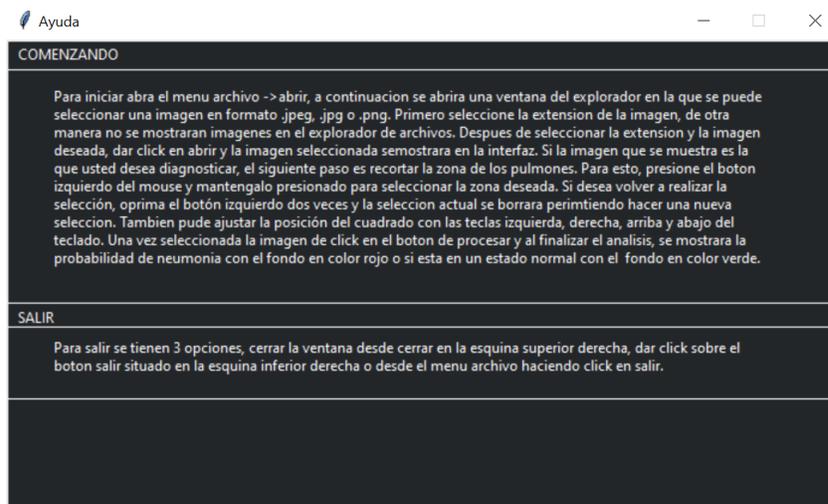


Figura 3.8: Ayuda.

El **Ejemplo de recorte** que se hace de manera manual para seleccionar el área a procesar y no incluir toda la imagen. Este recorte se hace seleccionando el área deseada con el ratón, el área seleccionada en cualquier caso se realiza en forma de cuadrado, esto es para que no se pierda la relación de aspecto de los pulmones ya que se hace una redimensión

de la parte seleccionada, se puede ajustar el cuadro de selección con las flechas del teclado (Fig.3.9).

Por último **Acerca de** que muestra las versiones de las herramientas utilizadas para crear la interfaz (Fig.3.10). Estas son: *python: 3.7.6*, *tkinter: 8.6*, *spyder: 4.0.1*.

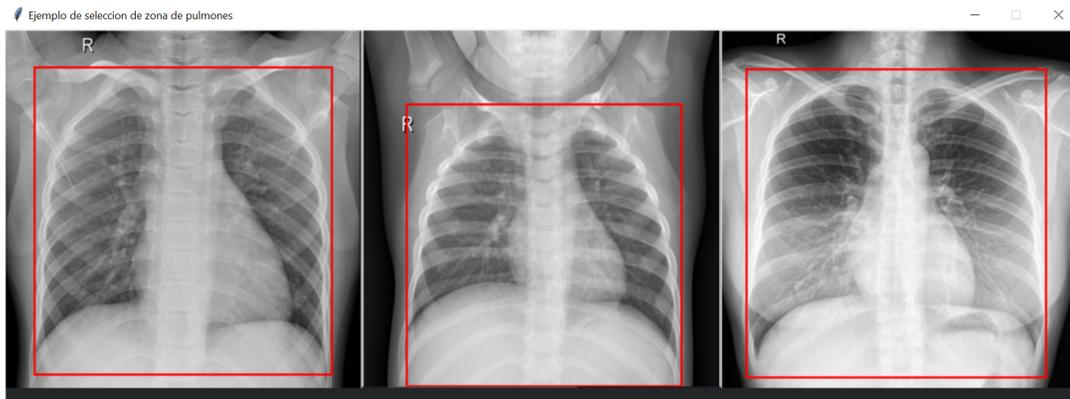


Figura 3.9: Ejemplo de recorte de radiografías.

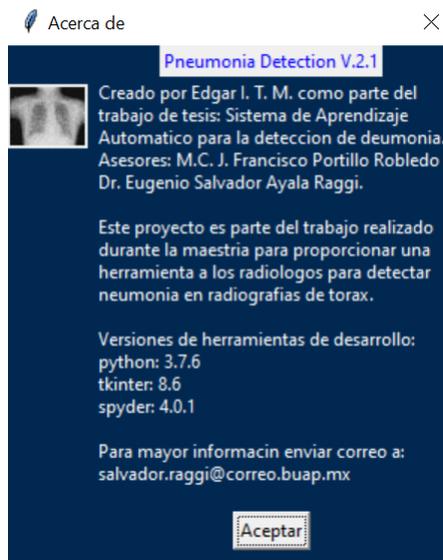


Figura 3.10: Acerca de.

Si no se carga una imagen a la interfaz y se oprime el botón de procesar, la pantalla adquiere un fondo naranja y se despliega un mensaje con el texto: «*Cargue una imagen*» (Fig.3.11).

De igual manera, sino se selecciona la sección de la imagen a procesar, se muestra una leyenda en la pantalla con el texto: «*Recorte la imagen*» y con el fondo en color naranja



Figura 3.11: Pantalla indicando cargar una imagen

como se muestra en la figura 3.12.



Figura 3.12: Pantalla indicando hacer el recorte

Capítulo 4

Resultados

En esta sección se presentan los resultados al aplicar el algoritmo k-NN propuesto. Para visualizar el comportamiento del algoritmo se utilizó un conjunto de datos creado de manera sintética con dos características de tal manera que fuera graficable para observar si se comportaba de la manera deseada y poder hacer modificaciones o correcciones. Posteriormente se presentan los resultados con el conjunto de datos de las radiografías de rayos-X que se descargaron de Kaggle, el conjunto de datos se llama *COVID-19 radiography database* [36]. Por último, se muestra el uso de la interfaz gráfica. Las pruebas presentadas en este capítulo se realizaron utilizando *python* como lenguaje de programación y *spyder* como entorno de desarrollo.

4.1. Conjunto de Datos Círculo

Las pruebas iniciales en el desarrollo del algoritmo se realizaron utilizando un conjunto de entrenamiento sintético, que es un grupo de puntos de dos características creados aleatoriamente con valores entre -1 y 1 en donde los puntos que se encuentran dentro de un círculo de radio 0.8 pertenecen a una clase y los que quedan fuera del círculo a otra clase. Estas pruebas se realizaron con el objetivo de observar el comportamiento del algoritmo ya que al ser datos de dos características se pueden graficar. A continuación, se muestran los resultados obtenidos con este conjunto de datos.

En la figura 4.1 observamos el conjunto de datos inicial, los puntos en color azul representan a una clase y los puntos de color rojo a otra. La figura 4.2 muestra el resultado del primer paso del algoritmo que realiza la poda, esta poda se realiza aplicando k-NN y tomando las muestras cuya fuerza de clasificación sea menor a un umbral.

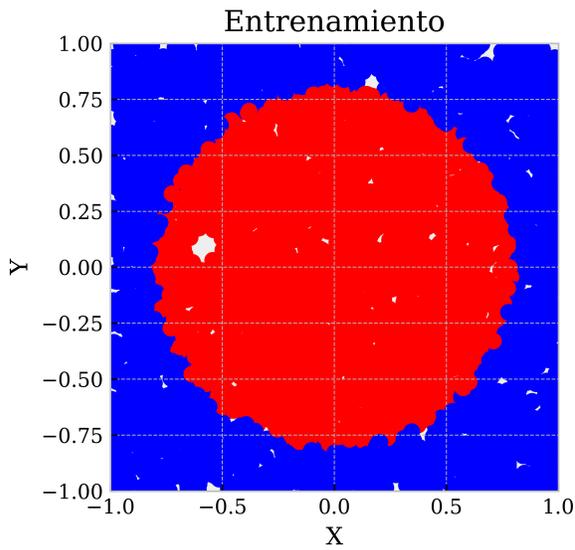


Figura 4.1: Conjunto de muestras inicial.

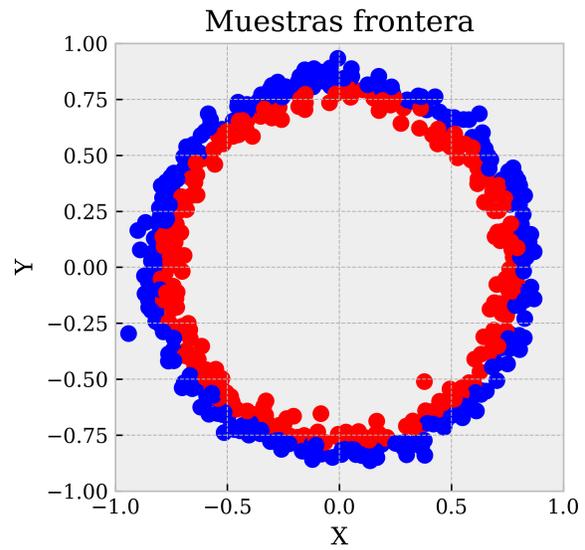


Figura 4.2: Conjunto de muestras podado.

El siguiente paso del algoritmo es seleccionar de manera aleatoria, un numero de prototipos iniciales de cada clase definidos por el usuario, en estas pruebas se tomaron 100 muestras por clase (200 muestras en total), este proceso se ilustra en la Fig. 4.3. Por último, en la Fig. 4.4 se muestran los prototipos resultantes del entrenamiento.

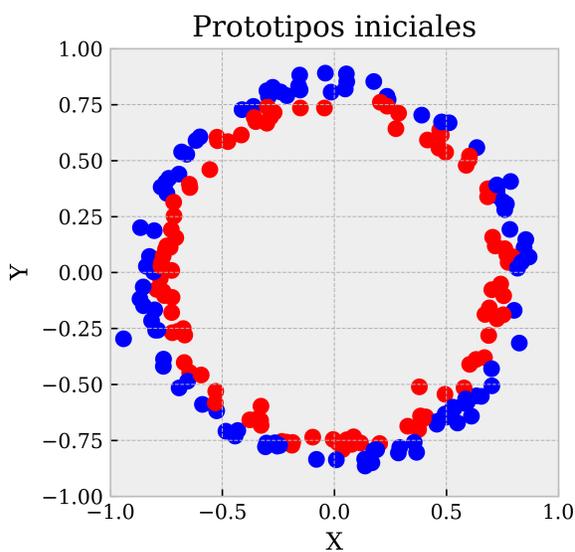


Figura 4.3: Prototipos iniciales.

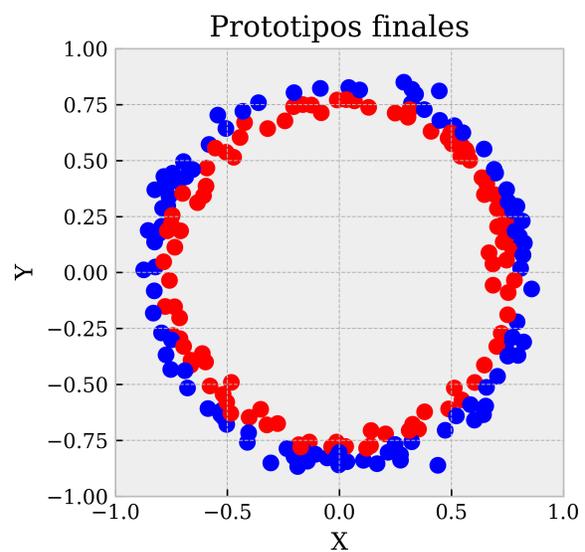


Figura 4.4: Prototipos finales.

La tasa de reconocimiento utilizando k-NN ponderado tradicional se muestra de manera gráfica en la Fig. 4.5. En color guinda se muestran los resultados utilizando todo el conjunto de entrenamiento, mientras que en color verde se muestra la tasa de reconocimiento utilizando 100 muestras de cada clase tomadas del conjunto original de manera aleatoria. Se utilizan valores de k del 1 al 10, se puede observar que no hay una variación considerable en los resultados y la exactitud es prácticamente constante.

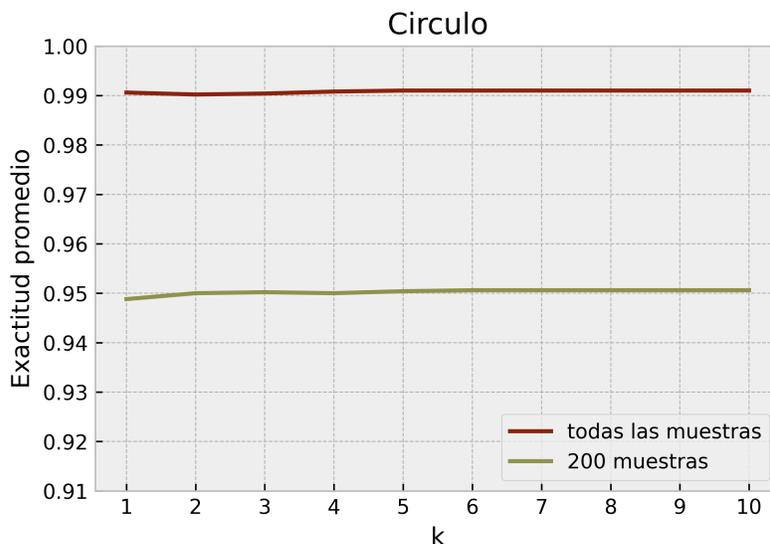


Figura 4.5: Tasa de reconocimiento k-NN tradicional utilizando todo el conjunto de entrenamiento y 200 muestras seleccionadas de manera aleatoria para diferentes valores de k.

Las figura 4.6 muestra el comportamiento del algoritmo propuesto en una prueba en la que se utiliza el conjunto de entrenamiento *círculo* y se incrementa el valor de k en 1 en cada entrenamiento. Inicialmente se tenía planteado eliminar prototipos durante el entrenamiento, si la memoria de fuerza de clasificación de las muestras dentro del prototipo, alcanzaban un umbral promedio de 0.99, esto, para dejar solo los prototipos más relevantes, pero al hacer esta prueba observamos que la tasa de reconocimiento es menor en comparación con la tasa de reconocimiento obtenida si no se eliminan prototipos.

Comparando los resultados de la figura 4.5 con la figura 4.6 podemos observar que algoritmo propuesto es capaz de obtener mejores resultados que una selección aleatoria del mismo número de prototipos y que alcanza una tasa de reconocimiento cercana a la obtenida al utilizar todo el conjunto de muestras. Lo que demuestra que el algoritmo es capaz de generalizar el conjunto de muestras con un número de prototipos mucho menor

a las muestras iniciales.

En la figura 4.7 realizamos la comparación entre reentrenamientos aumentando el valor de k después de cada entrenamiento y reentrenamientos manteniendo el valor de k constante, podemos observar que no hay diferencias significativas para este conjunto de datos. Excepto con $k=1$ y $k=10$.

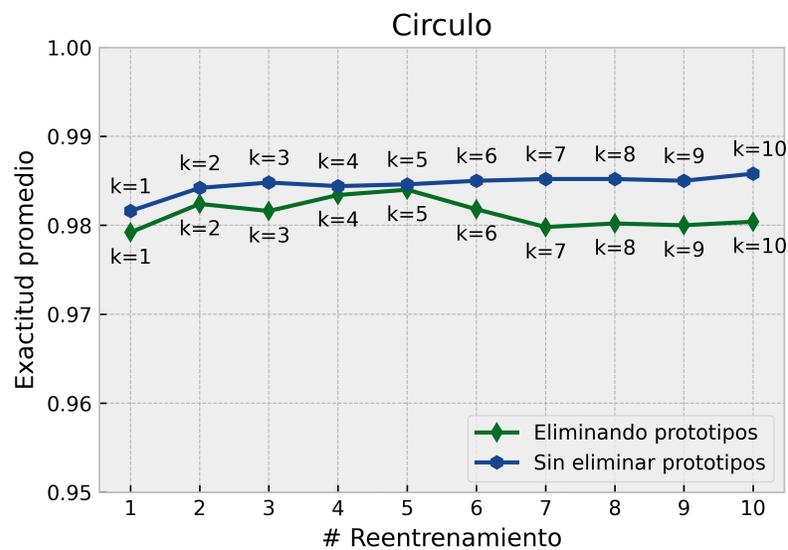


Figura 4.6: Comparación de reentrenamientos sin eliminar prototipos y eliminando prototipos.

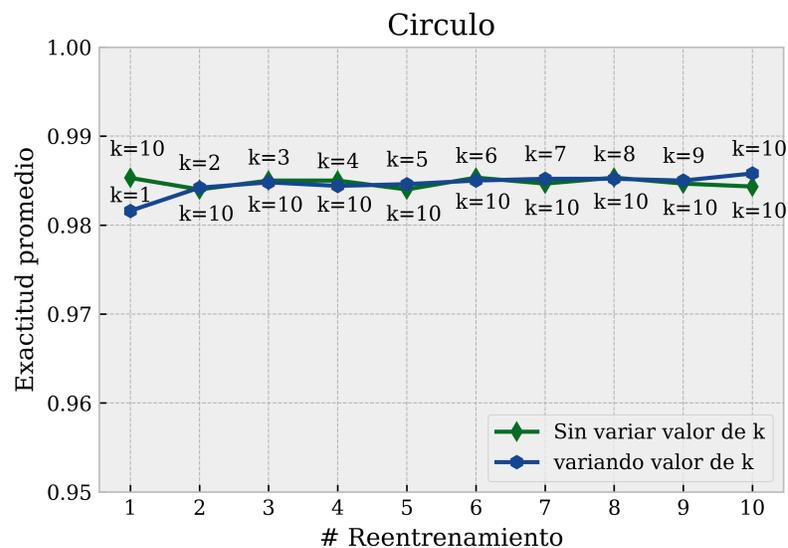


Figura 4.7: Comparación de reentrenamientos con k constante y variando el valor de k después de cada entrenamiento.

4.2. Conjunto de Datos Radiografías Rayos-X

Las pruebas siguientes se realizaron con el conjunto de datos disponible en Kaggle [36], que consta de 1341 imágenes de radiografías normales, 1345 con neumonía viral y 1200 con imágenes de COVID-19.

Una vez que se obtienen los conjuntos de entrenamiento y prueba, el siguiente paso es aplicar el método de eigenfaces sobre el conjunto de entrenamiento y proyectar el conjunto de prueba. Posteriormente se normaliza el conjunto de entrenamiento de manera que se tenga una desviación estándar de 1. La desviación estándar y los promedios obtenidos en esta normalización se utilizan para normalizar el conjunto de prueba. El siguiente paso es la selección de características más discriminantes del conjunto de entrenamiento y se toman las mismas del conjunto de prueba. Por último, hacemos una poda de las muestras de entrenamiento que al aplicarles k-NN ponderado tradicional se obtiene una fuerza de clasificación menor a 0.7 que son muestras que no pertenecen a la frontera. Estas muestras se utilizan para entrenar el algoritmo propuesto.

4.2.1. Imágenes sin preprocesar

Para esta prueba, utilizamos las carpetas completas de imágenes de neumonía y normales, la resolución de todas las imágenes es de 1024x1024, para nuestro algoritmo se redimensionaron a 256x256, sin aplicar algoritmos de mejora de contraste ni recortes. En los resultados se muestra el valor de *accuracy* como promedio de los resultados obtenidos con validación cruzada de 5 particiones y 5 iteraciones, en donde en cada iteración se utiliza una partición diferente como conjunto de prueba y las restantes como conjunto de entrenamiento.

En la Fig.4.8, la línea color arena muestra los resultados obtenidos con k-NN ponderado tradicional utilizando 100 muestras de cada clase tomadas de forma aleatoria del conjunto de entrenamiento mientras que la línea guinda indica los resultados obtenidos con k-NN ponderado tradicional utilizando todo el conjunto de entrenamiento. En el eje y se muestra el promedio de exactitud para diferentes valores de k (eje y).

En la Fig.4.9 podemos observar los resultados utilizando el algoritmo propuesto. Se seleccionaron inicialmente 100 muestras por clase de manera aleatoria del conjunto de entrenamiento podado. Podemos observar la exactitud promedio en cada reentrenamiento,



Figura 4.8: Exactitud promedio aplicando k-NN con diferentes valores de k. La línea arena pertenece a los resultados utilizando 100 muestras de manera aleatoria por clase del conjunto de entrenamiento podado. La línea guinda muestra los resultados utilizando todo el conjunto de entrenamiento.

en donde, además, se incrementó el valor de k (línea verde) comparado con un valor constante de k (línea azul). A diferencia del conjunto de datos *círculo*, se observa que en este caso si hay una diferencia significativa y se obtienen mejores resultados al variar el valor de k. Además, comparando la figura 4.8 con la línea azul de la figura 4.9 con un valor de k=10 se obtienen mejores resultados que utilizando todas las muestras.

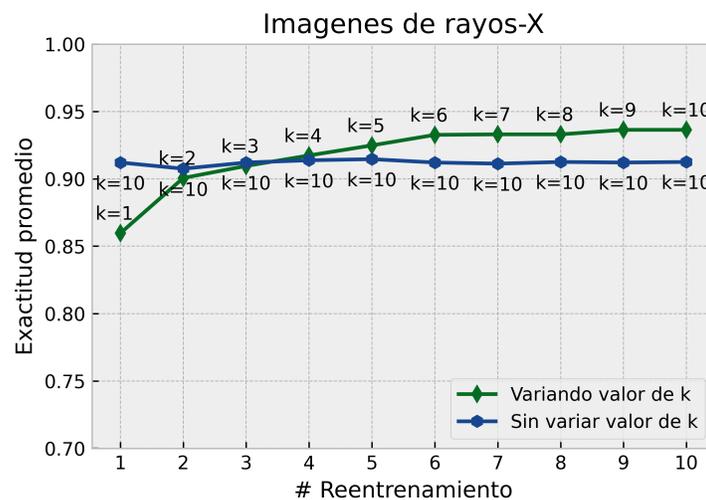


Figura 4.9: Exactitud promedio utilizando el algoritmo propuesto utilizando reentrenamientos y variando el valor de k en cada reentrenamiento.

En la tabla 4.1 aparecen los resultados máximos de exactitud utilizando 200 aleatorias muestras del conjunto de entrenamiento podado, todo el conjunto de entrenamiento

podado, estas dos aplicando k-NN ponderado y 200 prototipos del algoritmo propuesto reentrenado 10 veces e incrementando el valor de k en 1 después de cada entrenamiento. Podemos observar que, en este caso a diferencia del conjunto de datos del Círculo, la tasa de reconocimiento supera al k-NN tradicional.

Tabla 4.1: Exactitud promedio máxima utilizando 200 muestras, todo el conjunto de entrenamiento y 200 prototipos generados por el algoritmo propuesto mediante 10 reentrenamientos.

	200 muestras aleatorias	Todas las muestras	200 prototipos
Exactitud	86.10	91.70	93.63

4.2.2. Imágenes recortadas y con mejora de contraste

Esta prueba se realizó con el objetivo de seleccionar una estandarización en el contraste de las imágenes de manera que el sistema sea robusto a la variación de este para obtener una solución más general que no dependa del conjunto de entrenamiento y sea capaz de identificar neumonía en imágenes de otros conjuntos de datos.

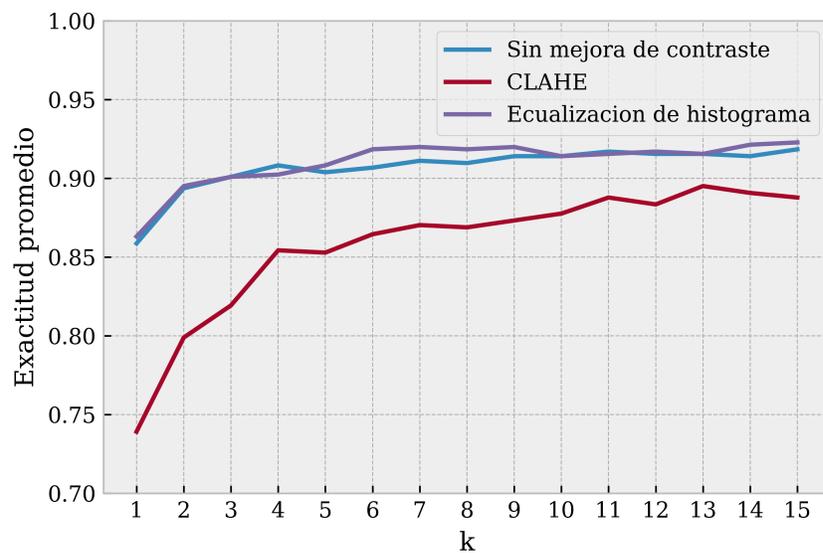


Figura 4.10: Comparación entre resultados utilizando imágenes recortadas: sin mejora de contraste, con mejora de contraste utilizando CLAHE, con mejora de contraste utilizando ecuilización del histograma.

La imagen 4.10 muestra los resultados en tasa de reconocimiento, utilizando 1000 imágenes seleccionadas de manera aleatoria del conjunto de datos para entrenamiento y el

resto para prueba. Las imágenes se recortaron de tal manera que se seccionara el área correspondiente a los pulmones, fue un recorte cuadrado sin pérdida de relación de aspecto. La línea guinda representa los resultados aplicando el algoritmo propuesto a las imágenes recortadas y aplicando ecualización de histograma adaptativa limitada por contraste (CLAHE). La línea morada corresponde a los resultados con las imágenes a las que se les aplicó ecualización del histograma como mejora de contraste. La línea azul corresponde a las imágenes sin aplicar mejora de contraste.

Aunque en la bibliografía mencionan que este método les produce mejores resultados, en nuestro caso, se puede apreciar que los peores resultados se obtienen aplicando CLAHE. La ecualización de histograma produce una leve mejora en comparación con las imágenes originales.

Las imágenes de las figuras 4.11, 4.12 y 4.13 muestran cuatro imágenes tomadas del conjunto de entrenamiento de manera aleatoria. Las primeras son las imágenes originales recortadas, las segundas son las mismas imágenes a las que se les aplicó CLAHE y las últimas son las mismas imágenes con ecualización del histograma. Visualmente las imágenes con CLAHE aplicado presentan mayor detalle.

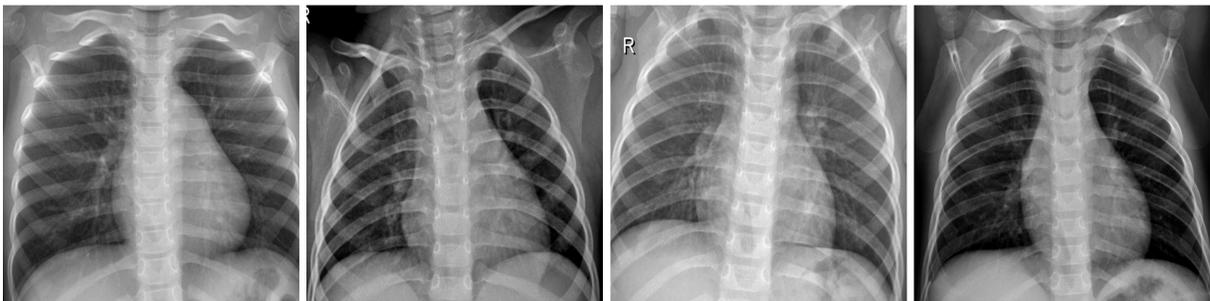


Figura 4.11: Imágenes originales recortadas.



Figura 4.12: Imágenes con CLAHE.

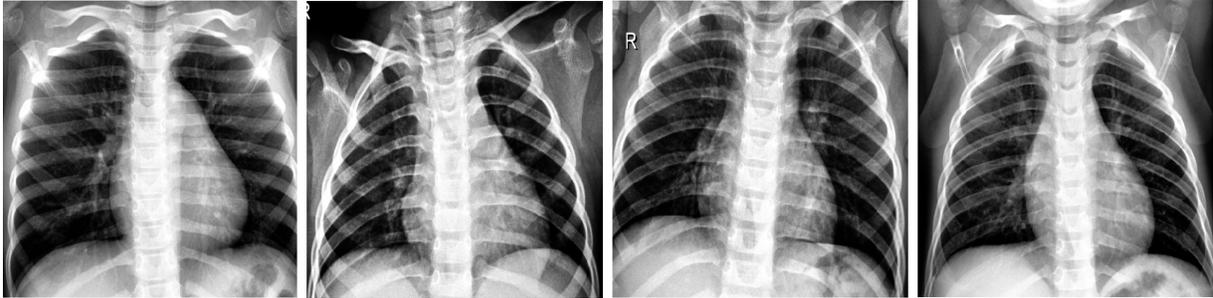


Figura 4.13: Imágenes con ecualización del histograma.

La tabla 4.2 muestra los valores máximos de exactitud alcanzados en los reentrenamientos, con las imágenes normales se alcanza un máximo de 91.69 con $k=11$, con las imágenes a las que se les aplicó CLAHE el mejor resultado se obtiene cuando $k=13$ y para las imágenes a las que se les aplicó ecualización del histograma logran un máximo de 92.27 con $k=15$. Aunque es notable la reducción de la exactitud de las imágenes recortadas en comparación con las imágenes sin recortar, el seleccionar el área del pulmón permite darle robustez al sistema, ya que algunas imágenes de otros bancos aparecen recortadas y no se puede crear información que no existe por lo que es mejor hacer el recorte para poder utilizar la interfaz en imágenes de otros bancos.

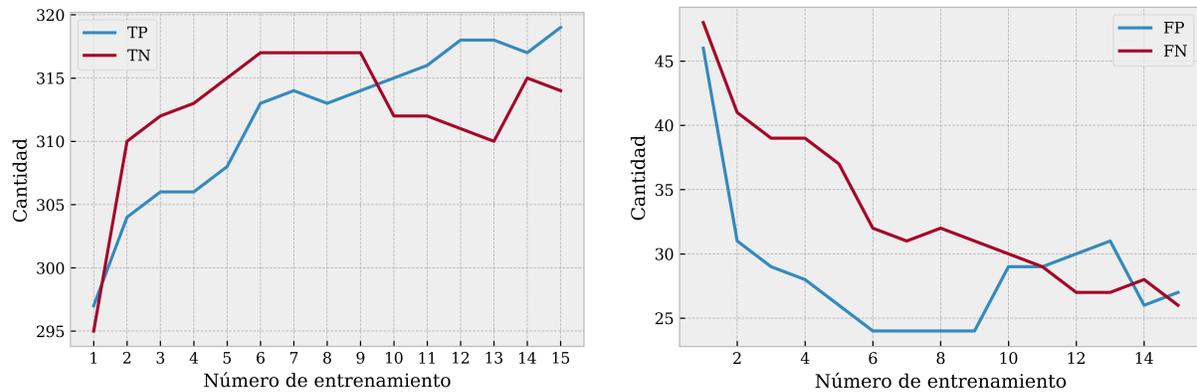
Tabla 4.2: Exactitud máxima para las imágenes originales, imágenes con CLAHE e imágenes con histograma ecualizado.

	Imágenes originales	CLAHE	Ecualización del histograma
Exactitud	91.69	89.50	92.27

En la siguiente tabla se muestra la matriz de confusión para las imágenes con ecualización del histograma aplicado y con 15 reentrenamientos. Los resultados completos de los entrenamientos pueden ser consultados en el apéndice A.

		Resultado de la predicción		
		Positivo	Negativo	
Valor Actual	Positivo	VP = 319	FN = 26	VP+FN = 345
	Negativo	FP = 27	VN = 314	FP+VN = 341

Cabe destacar que tanto los verdaderos positivos como los verdaderos negativos se detectan en cantidades similares por lo que tanto la sensibilidad como la especificidad presentan valores cercanos. La métrica de sensibilidad muestra la cantidad de verdaderos positivos que el modelo ha clasificado en función del número total de valores positivos.



(a) Comportamiento de los verdaderos positivos(TP) y verdaderos negativos(TN) durante el entrenamiento

(b) Comportamiento de los verdaderos positivos(TP) y verdaderos negativos(TN) durante el entrenamiento

Figura 4.14: Desempeño del algoritmo propuesto durante el entrenamiento.

También se conoce como *recall*. Una sensibilidad alta significa pocos falsos negativos.

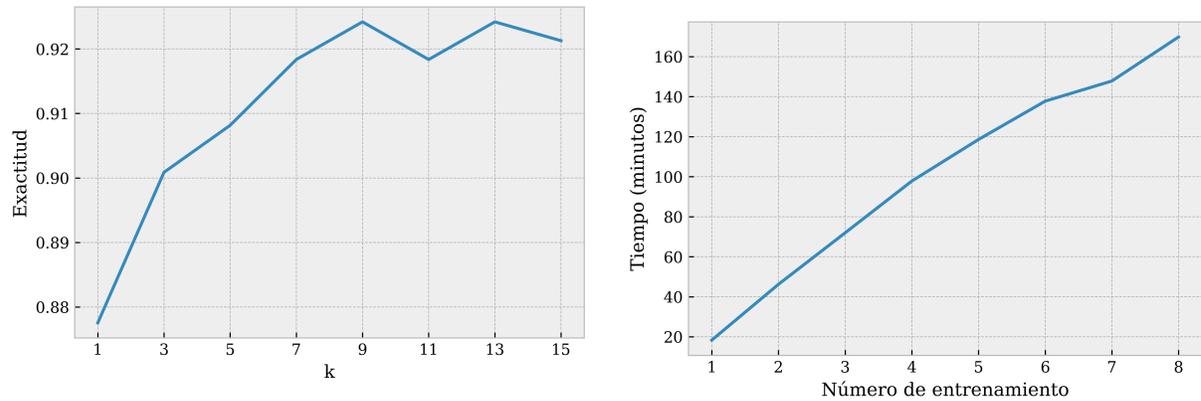
$$Sensibilidad = \frac{VP}{VP + FN} = \frac{319}{319 + 26} = 0.9246$$

La especificidad es la fracción de verdaderos negativos, indica la capacidad de nuestro estimador para discriminar los casos positivos, de los negativos. Se trata de los casos negativos que el algoritmo ha clasificado correctamente.

$$Especificidad = \frac{VN}{FP + VN} = \frac{314}{314 + 27} = 0.9208$$

La figura 4.14 muestra la etapa de entrenamiento donde se observa que tanto el número de verdaderos positivos como el número de verdaderos negativos aumenta (Fig.4.14a) y el número de falsos positivos y falsos negativos disminuye (Fig.4.14b).

En busca de hacer más ligero el entrenamiento, una de las ideas que surgió fue la de en lugar de incrementar el valor de k en 1, que el incremento sea en 2, de esta manera en el rango de k [1,15] se tienen 8 entrenamientos en lugar de 15. Los resultados de esta prueba se observan en la figura 4.15a . Se puede observar que al igual que los incrementos en 1, el algoritmo mejora con cada reentrenamiento, alcanzando valores de exactitud similares y esto en menos tiempo. De esta manera el tiempo total del entrenamiento se reduce al igual que el tiempo máximo de un solo entrenamiento como se observa en la figura 4.15b. En el apéndice A se muestra la tabla de entrenamiento completa.

(a) Exactitud en reentrenamientos incrementado el valor de k en 2

(b) Tiempo de ejecución de cada reentrenamiento

Figura 4.15: Resultados de reentrenamientos incrementando el valor de k en 2.

La tabla 4.3 muestra los valores de exactitud máximos alcanzados en las imágenes de entrenamiento recortadas y con ecualización del histograma. Aplicando el k -NN tradicional se obtiene un máximo con $k=15$ de 91.10, con reentrenamientos e incrementando el valor de k en 1 después de cada entrenamiento, el máximo es de 92.27 con $k=15$. Con el algoritmo propuesto y reentrenamientos incrementando el valor de k en 2 se obtienen máximos con $k=9$ y $k=13$.

Tabla 4.3: Exactitud promedio máxima utilizando todo el conjunto de entrenamiento con k -NN tradicional, reentrenamientos incrementando el valor de k en 1 y reentrenamientos incrementando el valor de k en 2 con algoritmo propuesto.

	k -NN tradicional	Alg. propuesto $k + 1$	Alg. propuesto $k + 2$
Exactitud	91.10	92.27	92.42

4.3. Interfaz Gráfica

La interfaz gráfica aplica los mismos procesos que la etapa de prueba al procesar una imagen de entrada que se desea clasificar. Para procesar una imagen primero se debe de cargar una imagen en la interfaz (Fig.4.16). Después se realiza un recorte cuadrado que se hace de manera manual, la selección es cuadrada debido a que todas las imágenes se redimensionan a 256×256 y de esta manera no se pierde la relación de aspecto (Fig.4.17). Antes de la redimensión se aplica mejora de contraste. La imagen redimensionada se proyecta en el eigenspacio generado durante el entrenamiento tomando en cuenta la imagen promedio y los eigenfaces. Por último, se aplica la normalización y se toman las características más

relevantes que se seleccionaron durante el entrenamiento.

Una vez seleccionada el área en la imagen de entrada, se oprime el botón de procesar y de manera casi instantánea se obtiene el resultado del procesamiento. Si no se detectó neumonía, se muestra un mensaje en la pantalla con el fondo verde y el texto: «Normal» (Fig.4.18), por el contrario, si se detectó neumonía en la imagen, el fondo de la pantalla cambia a rojo y se muestra la probabilidad de neumonía(4.20).



Figura 4.16: Ejemplo 1 imagen a procesar.



Figura 4.17: Ejemplo 1 imagen recortada.

En las figuras 4.19, 4.20 y 4.21 se muestra otro ejemplo para una imagen con neumonía. El resultado de la predicción del sistema se muestra de manera correcta en la pantalla de la interfaz.



Figura 4.18: Ejemplo 1 resultado sin neumonía.



Figura 4.19: Ejemplo 2 imagen a procesar.

4.4. Discusión de Resultados

Pruebas de versiones anteriores del algoritmo pueden ser consultadas en el apéndice B. También realizamos una publicación en el II Congreso Internacional y X Congreso Nacional de Ciencias de la Computación CONACIC2020, la portada del artículo aparece en el apéndice C.

El comportamiento del algoritmo propuesto entre distintos bancos de datos varía significativamente, como ejemplo tenemos los dos bancos analizados: el circulo y las radiografías de neumonía. En el primero los reentrenamientos variando el valor de k y sin variar el valor de k no representan una variación significativa mientras que en el segundo la diferencia es clara y existe una mejora cuando se incrementa el valor de k en cada reentrenamiento. Esto se debe a que la distribución de los datos es muy distinta al igual que el número de



Figura 4.20: Ejemplo 2 imagen recortada.



Figura 4.21: Ejemplo 2 resultado con neumonía.

características que cada uno contiene.

Esto no es propio únicamente de este algoritmo, sino que la mayoría de ellos presenta este comportamiento, por ejemplo, para la mejora de contraste, en la literatura el método más usado es la ecualización del histograma adaptativa limitada por contraste (CLAHE) y mediante las pruebas presentadas es claro que para este sistema no es benéfico.

Los resultados de la mejora de contraste pueden ser explicados con la imagen 4.22 en donde se gráfica la suma acumulada de la variación total de las imágenes entrenamiento. La matriz de covarianza genera varios eigenfaces, en la primera figura 4.22a se presentan los eigenfaces necesarios para alcanzar el 98 % de la variación total en imágenes a las que se les aplicó CLAHE. En la figura 4.22b se presenta el mismo gráfico, pero con imágenes a las que se les aplicó ecualización del histograma. Podemos observar que se necesita un

mayor número de eigenfaces para representar la variación total en el conjunto de imágenes con CLAHE. Esto quiere decir que hay más variación entre ellas.

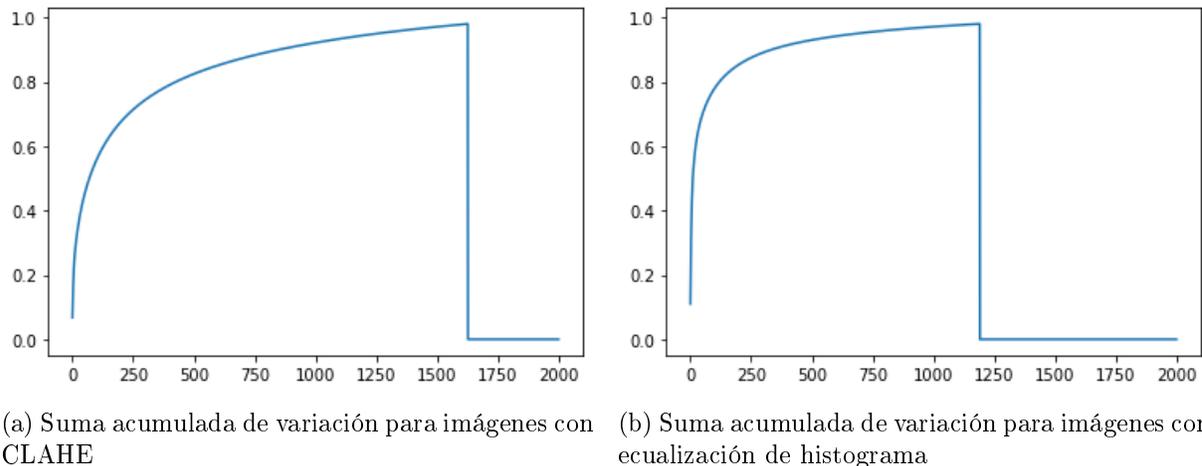


Figura 4.22: Variación de las imágenes de entrenamiento. En el eje X aparecen el número de eigenvalores ordenados de mayor a menor, en el eje Y se muestra la variación. El corte ocurre cuando se alcanza el 98 % de la variación total.

El método original de eigenfaces sugiere utilizar los eigenfaces con los eigenvalores más altos, como este algoritmo no es supervisado y más bien es un algoritmo de reducción de dimensión, estos eigenfaces no son los mejores para clasificación, es por esto por lo que se optó por implementar un método de selección de características basado en Fisherfaces que toma en cuenta la distribución de las clases.

Durante el entrenamiento, con cada reentrenamiento al algoritmo se vuelve más tardado ya que se van llenando las memorias de los prototipos y la función *mem_label_estimation* aplica k-NN a todas ellas. El tamaño máximo de la memoria de muestras de cada prototipo es de 160, teniendo 200 prototipos, el número máximo de muestras almacenadas es de 32,000. En la figura 4.23 se muestra este comportamiento, en el eje X tenemos el número de reentrenamiento y el eje Y el tiempo de ejecución en minutos.

A modo de comparación citamos el trabajo realizado en [65] presenta resultados semejantes en términos de exactitud. Utilizan características *image-based* y *LBP image-based*, con las primeras obtienen una exactitud de 92.60 %, con las segundas 92.44 % y combinando ambas alcanzan el 94.19 %, utilizando como clasificador *support vector machines* (SVM). Utilizan el mismo conjunto de datos y el mismo número de imágenes en sus pruebas. Los resultados obtenidos en el presente trabajo compiten con este y otras soluciones

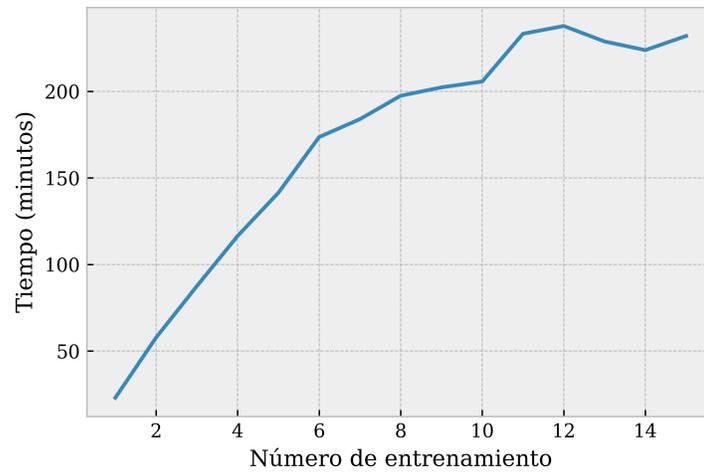


Figura 4.23: Tiempos de ejecución.

propuestas en la literatura.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1. Conclusiones

Durante este trabajo se evaluaron diferentes técnicas de extracción de características. Las características de textura de Haralick de igual manera no presentan buenos resultados, creemos que para que estas características funcionen, se tiene que seccionar la región de interés y hacer una selección de las características que más contribuyen a la clasificación para eliminar las que no aportan información discriminativa. Las bolsas de palabra no produjeron los resultados esperados, al parecer porque, los puntos de interés se crean en las partes del pulmón que no se ve afectado por la neumonía, es decir no es sensible al área de la neumonía sino a la ausencia de esta, aunque se hicieron pruebas con diferentes valores de sigma para detectar puntos de interés más grandes, era difícil para el algoritmo detectar las estructuras de consolidación.

Los mejores resultados los obtuvimos con el algoritmo propuesto, utilizando como vector de características los pesos generados por el algoritmo de eigenfaces, aplicando normalización y selección de características además de realizar reentrenamientos variando el valor de k en cada entrenamiento y utilizando como prototipos los generados en el entrenamiento anterior.

En la literatura, la mayoría de los trabajos mejoran el contraste utilizando la ecualización de histograma adaptativa limitada por contraste (CLAHE), en nuestra propuesta se comprobó que obtuvimos mejores resultados con la ecualización del histograma.

Para la selección de características además de método propuesto se consideró aplicar Fisherfaces, el problema es que, con este método, hay como máximo $c - 1$ valores propios

generalizados distintos de cero, donde c es el número de clases, para el algoritmo de clasificación propuesto basado en k -NN no es benéfico porque al ser solo dos clases únicamente queda una característica, esto le quita libertad de movimiento al algoritmo.

El algoritmo inicialmente se había diseñado de tal manera que, si una nueva muestra no podía ser clasificada correctamente mediante los prototipos existentes, la muestra se creaba como un nuevo prototipo, además, si el promedio de las muestras almacenadas en el prototipo tenía un promedio mayor a un umbral de .99 estas muestras se eliminaban ya que con un valor CLS tan alto, estas muestras no pertenecen a la frontera. Sin embargo, al realizar la comparación entre reentrenamientos eliminando prototipos y no eliminando, muestra mejores resultados sino se eliminan.

Asimismo, versiones anteriores del algoritmo estudiaron crear nuevos prototipos con las muestras que no pudieran ser correctamente clasificadas con los prototipos presentes ni con el acercamiento de ellos. El problema de este comportamiento es que no se tiene un control sobre los prototipos creados y se puede sobre ajustar el sistema a las muestras de entrenamiento. La versión final mejora este comportamiento al definir el número de prototipos a sintetizar.

La complejidad del algoritmo radica únicamente en el entrenamiento, ya que para la clasificación se aplica k -NN tradicional ponderado tomando como prototipos los prototipos obtenidos durante la etapa de entrenamiento y que están ubicados de tal manera que generalizan el conjunto de entrenamiento.

Aunque se aplicaron procesos a las imágenes de tal manera que se estandarizaran lo mejor posible, tales como el recorte y la ecualización del histograma, no fue posible generalizar el sistema de tal manera que los prototipos generados en un banco funcionaran en otros bancos. Esto se debe a la variación de las imágenes y a que se necesita una correcta alineación, escalado y centrado del área de los pulmones para tener un sistema más robusto.

La principal aportación de este trabajo es la contribución al campo de aprendizaje automático de un algoritmo de generación/selección de instancias, a partir, de muestras reales de un conjunto de entrenamiento, para aplicaciones de clasificación. Este algoritmo está basado en k -NN (k vecinos más cercanos) y tiene como objetivo obtener un conjunto reducido de prototipos que represente de mejor manera al conjunto de datos original. Con dos conjuntos de datos se muestra el comportamiento de este algoritmo, para el conjunto

de datos de radiografías de rayos-X, los prototipos generados producen un mejor resultado que utilizando todo el conjunto de entrenamiento.

5.2. Trabajo Futuro

Como PCA es sensible a la escala, alineación y posición de las imágenes, creemos que podemos mejorar los resultados presentados, alineando y escalando las imágenes para obtener la menor variación posible. Para esto es necesario identificar puntos en las imágenes que sean fáciles de identificar en todas las radiografías.

Una vez identificados los puntos de referencia es posible hacer un recorte del área de los pulmones más exacta y de manera automática para eliminar las variaciones que se puedan presentar por un mal recorte realizado de manera manual.

Aunque el algoritmo propuesto es de aprendizaje incremental, esto no es del todo aprovechado ya que el algoritmo de PCA no es incremental y las nuevas imágenes se proyectan sobre el mismo eigenspacio. Una posible mejora puede ser el elegir otro tipo de características para que todo el sistema sea de aprendizaje incremental.

El algoritmo propuesto es computacionalmente intensivo, como trabajo futuro se plantea realizar el análisis de complejidad y el análisis de tiempos de ejecución, además de hacer adecuaciones para hacer más ligera la etapa de entrenamiento y generar un algoritmo optimizado.

Por último, se plantea realizar pruebas con otras bases de datos y realizar comparaciones con otros trabajos.

Apéndice A

Tablas de Reentrenamientos

En este apéndice se encuentran las tablas completas con los resultados de cada entrenamiento para las imágenes recortadas sin mejora de contraste y con mejora de contraste. Las tablas muestran los verdaderos positivos (TP), verdaderos negativos (TN), falsos negativos (FN) y falsos positivos (FP), también se muestran los tiempos de ejecución, el número de prototipos y la exactitud.

Tabla A.1: Reentrenamiento con el algoritmo propuesto e imágenes originales recortadas.

k	TP	TN	FP	FN	Tiempo(min)	No.P.	Exactitud
1	299	290	51	46	27.58	200	0.858601
2	311	302	39	34	82.22	200	0.893586
3	321	297	44	24	131.01	200	0.900875
4	316	307	34	29	167.21	200	0.908163
5	311	309	32	34	191.88	200	0.903790
6	309	313	28	36	200.69	200	0.906706
7	313	312	29	32	220.02	200	0.911079
8	314	310	31	31	214.13	200	0.909621
9	313	314	27	32	210.16	200	0.913994
10	313	314	27	32	215.25	200	0.913994
11	313	316	25	32	218.79	200	0.916910
12	313	315	26	32	216.44	200	0.915452
13	314	314	27	31	230.27	200	0.915452
14	314	313	28	31	229.54	200	0.913994
15	314	316	25	31	239.80	200	0.918367

Tabla A.2: Reentrenamiento con el algoritmo propuesto e imágenes recortadas y CLAHE aplicado.

k	TP	TN	FP	FN	Tiempo(min)	No.P.	Exactitud
1	286	221	120	59	36.82	200	0.739067
2	288	260	81	57	119.19	200	0.798834
3	295	267	74	50	189.41	200	0.819242
4	297	289	52	48	232.70	200	0.854227
5	296	289	52	49	260.70	200	0.852770
6	298	295	46	47	292.48	200	0.864431
7	304	293	48	41	288.71	200	0.870262
8	310	286	55	35	299.08	200	0.868805
9	309	290	51	36	307.98	200	0.873178
10	309	293	48	36	302.58	200	0.877551
11	310	299	42	35	321.40	200	0.887755
12	308	298	43	37	310.46	200	0.883382
13	311	303	38	34	342.52	200	0.895044
14	313	298	43	32	363.98	200	0.890671
15	308	301	40	37	370.16	200	0.887755

Tabla A.3: Reentrenamiento con el algoritmo propuesto e imágenes recortadas con ecualización del histograma.

k	TP	TN	FP	FN	Tiempo(min)	No.P.	Exactitud
1	297	295	46	48	23.02	200	0.862974
2	304	310	31	41	57.79	200	0.895044
3	306	312	29	39	87.51	200	0.900875
4	306	313	28	39	116.48	200	0.902332
5	308	315	26	37	141.44	200	0.908163
6	313	317	24	32	173.61	200	0.918367
7	314	317	24	31	183.96	200	0.919825
8	313	317	24	32	197.49	200	0.918367
9	314	317	24	31	202.33	200	0.919825
10	315	312	29	30	205.69	200	0.913994
11	316	312	29	29	233.30	200	0.915452
12	318	311	30	27	237.78	200	0.916910
13	318	310	31	27	228.85	200	0.915452
14	317	315	26	28	223.83	200	0.921283
15	319	314	27	26	232.00	200	0.922741

Tabla A.4: Reentrenamientos incrementando el valor de k en 2, con el algoritmo propuesto e imágenes recortadas con ecualización del histograma.

No.Ent	k	TP	TN	FP	FN	Tiempo(min)	Exactitud
1	1	297	305	36	48	18.32	0.87755
2	3	304	314	27	41	46.21	0.90087
3	5	307	316	25	38	71.95	0.90816
4	7	310	320	21	35	97.81	0.91837
5	9	315	319	22	30	118.60	0.92420
6	11	313	317	24	32	137.77	0.91837
7	13	316	318	23	29	147.83	0.92420
8	15	310	322	19	35	169.84	0.92128

Apéndice B

Pruebas Previas

En este apéndice se muestran resultados previos a la versión final del algoritmo, además se utiliza un banco de datos diferente y otros métodos de extracción de características. Las pruebas de esta sección se realizaron con el conjunto de datos de mendeley [34], tomando únicamente 1000 imágenes de neumonía y 1000 imágenes normales de manera aleatoria, a estas se les aplicó CLAHE como preprocesamiento, utilizando *python* como lenguaje de programación y *spyder* como entorno de desarrollo.

Bolsas de palabras

Mediante la utilización de bolsas de palabra aplicadas a imágenes, se realizaron pruebas para evaluar la viabilidad de utilizar esta técnica. El proceso se describe a continuación:

Lo primero que hacemos es extraer los descriptores de SIFT para cada imagen tomando como parámetros los propuestos por D. Lowe. El número de capas por octava es de 3, umbral de intensidad de 0.04, sigma inicial de 1.6, umbral de borde de 5.

El siguiente paso es encontrar los centroides por clase, es decir obtenemos los descriptores para las imágenes de neumonía y obtenemos con k-means 30 centroides. Realizamos lo mismo con las imágenes de personas sin neumonía. Al final tendremos 30 centroides por cada clase, 60 en total.

El siguiente paso es hacer la proyección de los puntos para cada centroide y contabilizar el número de ocurrencias por centroide. De esta manera formamos una especie de histograma de 60 bins que utilizamos como vector de características. Los resultados se presentan más adelante en la sección de comparación de resultados.

Eigenfaces

Se utilizó la técnica de eigenfaces propuesta por M. Turk and A. Pentland, para obtener los pesos para poder reconstruir o representar las imágenes por medio de la combinación lineal resultante de la proyección. Estos pesos que describen a cada imagen se utilizaron como vectores de características para entrenar un perceptrón de una sola capa intermedia y una capa de salida, también se hicieron pruebas aplicando el algoritmo propuesto y con kNN ponderado tradicional.

De manera empírica se notó que al realizar un entrenamiento aumentando el valor de k y utilizando las mismas muestras del primer entrenamiento, aumentaba considerablemente la exactitud del sistema. Es decir, se hace un entrenamiento tomando dos prototipos iniciales validos con un valor de $k=1$, como resultado obtenemos cierto número de prototipos que se utilizan como prototipos iniciales para un segundo entrenamiento con las mismas muestras de entrenamiento, este proceso se repite una vez más y al final se obtienen los prototipos finales que se utilizan para la validación.

Fisherfaces

Utilizamos el método de pca seguido por el discriminante de fisher, como solo tenemos dos clases, al aplicar LDA, se obtuvo que la clasificación resulta en términos de un umbral ya que la dimensión resultante después de LDA es de 1.

Comparación de resultados

La tabla B.5 muestra la comparación de resultados entre la clasificación otorgada por un perceptrón de una sola capa oculta variando el número de neuronas de esta y 20 épocas, un clasificador kNN con ponderación gaussiana con k vecinos por clase y el clasificador propuesto con k vecinos por clase para clasificar píxeles de la imagen y pesos otorgados aplicando PCA. Como conjunto de entrenamiento se utilizaron 1000 imágenes por clase y para prueba las carpetas completas de neumonía y normales que constan de 390 y 234 imágenes respectivamente.

Para las pruebas con LDA, obtuvimos 84.46 con fisherfaces y 76.00 con bolsas de palabra.

En la tabla B.2, podemos observar los resultados obtenidos al realizar reentrenamientos, en donde después de cada entrenamiento se varía el valor de k y se utilizan los prototipos

Tabla B.1: Comparación de la tasa de reconocimiento obtenida con las imágenes originales y con los pesos PCA

Píxeles				W_{pca}					
MLP		KNN		MLP		KNN		KNN INC	
Neuronas	ACC.	k	ACC.	Neuronas	ACC.	k	ACC.	k	ACC.
1	83.65	1	80.28	1	76.92	1	85.09	1	81.41
2	83.81	2	80.92	2	86.53	2	85.09	2	80.76
3	83.65	3	81.41	3	86.53	3	85.41	3	79.64
4	84.29	4	80.92	4	86.05	4	84.13	4	82.05
5	83.81	5	80.76	5	85.41	5	83.97	5	83.33
6	84.13	6	80.44	6	86.21	6	83.81	6	83.97
7	84.13	7	80.76	7	86.21	7	83.65	7	83.01
8	84.13	8	80.92	8	86.53	8	83.81	8	83.17
9	84.29	9	80.92	9	86.21	9	83.97	9	85.09
10	84.13	10	80.92	10	86.21	10	84.13	10	85.09

finales del entrenamiento anterior. Se muestra el número de entrenamiento (No. Ent.), número de vecinos por clase (k), número de prototipos resultantes del entrenamiento (Prot. Fin.), tiempo de entrenamiento (Tiempo) y la exactitud (ACC.), se revisaron 2 valores de sigma diferente, tomando como sigma la distancia del más cercano y dividida entre 2 ($d[0]/2$) y tomando como sigma la distancia del más lejano ($d[2*k]$).

Tabla B.2: Resultados de reentrenamiento variando el valor de k

Ent.	k	$\sigma d[2*k]$			$\sigma d[0]/2$		
		Prot. Fin.	Tpo.(min.)	ACC.	Prot. Fin.	Tpo.(min.)	ACC.
1	1	165	70	81.41	165	70	81.41
2	2	179	99	84.13	179	88	84.45
3	3	186	98	86.69	184	65	86.85
4	4	188	98	86.05	184	47	87.66
5	5	190	107	86.53	184	47	86.21
6	6	195	129	85.57	184	33	87.50
7	7	199	143	86.37	184	35	87.50
8	8	202	142	86.21	186	50	88.30
9	9	203	136	85.05	188	37	87.82
10	10	203	177	85.89	189	30	87.01

La tabla B.3 muestra los resultados al variar el valor de k después de cada entrenamiento y mezclar los datos de entrenamiento.

Tabla B.3: Resultados de reentrenamiento variando el valor de k y mezclando las muestras después de cada entrenamiento

Ent.	k	Prot. Fin.	Tpo.(min.)	ACC.
1	1	165	70	81.41
2	2	181	93	84.13
3	3	183	61	86.37
4	4	184	42	86.53
5	5	184	56	86.21
6	6	184	51	86.21
7	7	184	43	87.01
8	8	184	42	86.37
9	9	184	35	86.85
10	10	184	48	87.01

Tabla B.4: Resultados de reentrenamiento variando el valor de k, con datos sin normalizar y normalizados con 49 características seleccionadas

Ent.	k	Datos sin normalizar			Datos Normalizados		
		Tpo.(min.)	Prot. Fin.	ACC.	Tpo.(min.)	Prot. Fin.	ACC.
1	1	3.69	111	76.60	5.39	173	82.85
2	2	6.37	129	81.57	8.41	202	85.25
3	3	4.89	135	82.53	3.82	202	87.01
4	4	4.40	137	82.69	2.93	203	87.01
5	5	3.36	138	82.85	2.63	204	88.30
6	6	3.77	138	84.29	2.36	205	88.14
7	7	2.73	138	85.89	2.28	205	88.30
8	8	2.79	139	85.41	1.78	205	88.78
9	9	1.90	139	86.05	1.48	205	88.78
10	10	1.99	139	84.93	1.64	205	88.94
11	11	2.08	139	84.93	1.74	205	89.10
12	12	1.72	139	85.09	1.11	205	88.94
13	13	1.81	139	85.25	1.50	205	89.26
14	14	1.97	139	84.93	1.57	205	88.94
15	15	1.36	139	85.41	0.60	205	89.26

Tabla B.5: Comparación de 49 características seleccionadas contra las primeras 49 características otorgadas por PCA. Ambos conjuntos de datos se normalizaron.

49 características seleccionadas					primeras 49 características				
KNN Norm.		KNN INC.			KNN Norm.		KNN INC		
k	ACC.	Ent.	k	ACC.	k	ACC.	Ent.	k	ACC.
1	88.62	1	1	82.85	1	84.93	1	1	79.64
2	88.30	2	2	85.25	2	84.45	2	2	84.61
3	88.78	3	3	87.01	3	85.89	3	3	85.73
4	88.62	4	4	87.01	4	86.53	4	4	84.93
5	89.10	5	5	88.30	5	86.69	5	5	85.41
6	88.78	6	6	88.14	6	86.53	6	6	86.53
7	89.10	7	7	88.30	7	86.05	7	7	86.21
8	89.26	8	8	88.78	8	86.05	8	8	86.69
9	89.26	9	9	88.78	9	86.53	9	9	87.17
10	89.26	10	10	88.94	10	86.69	10	10	87.66
11	89.26	11	11	89.10	11	86.69	11	11	87.17
12	89.42	12	12	88.94	12	86.69	12	12	87.66
13	89.42	13	13	89.26	13	86.53	13	13	87.17
14	89.58	14	14	88.94	14	86.37	14	14	87.50
15	89.42	15	15	89.26	15	86.53	15	15	87.66

Apéndice C

Publicaciones

Una versión anterior del algoritmo se presentó en un artículo titulado: “Tras la síntesis de un conjunto reducido de prototipos óptimos para un algoritmo de clasificación: el k-NN evolutivo” en el II Congreso Internacional y X Congreso Nacional de Ciencias de la Computación CONACIC2020, del 9 al 13 de noviembre de 2020. El libro que contiene las publicaciones lleva el nombre “DESARROLLOS CIENTÍFICOS Y TECNOLÓGICOS EN LAS CIENCIAS COMPUTACIONALES”. Primera edición: 2020. ISBN: 978-607-8728-36-

7



BUAP

La Benemérita Universidad Autónoma de Puebla

A través de la Facultad de Ciencias de la Computación
otorga la presente

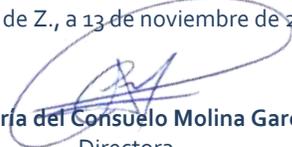
CONSTANCIA

A: Edgar Ibis Tacuapan Moctezuma, Salvador Eugenio Ayala Raggi,
Aldrin Barreto Flores, José Francisco Portillo Robledo

Por presentar el artículo, "*Tras la sintetización de un conjunto reducido de prototipos óptimos para un algoritmo de clasificación: el k-NN evolutivo*" en el II Congreso Internacional y X Congreso Nacional de Ciencias de la Computación CONACIC2020, del 9 al 13 de noviembre de 2020.

"Pensar bien, para vivir mejor"

Puebla de Z., a 13 de noviembre de 2020.


M.I. María del Consuelo Molina García
Directora.

Tras la sintetización de un conjunto reducido de prototipos óptimos para un algoritmo de clasificación: el k-NN evolutivo.

Edgar Ibis Tacuapan-Moctezuma¹, Salvador E. Ayala-Raggi¹, Aldrin Barreto-Flores¹,
José Francisco Portillo Robledo¹

¹ Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, Av. San Claudio y 18 sur, Col. Jardines de San Manuel, C.P. 72570, Puebla, Puebla, México. edgar.tacuapanmoc@alumno.buap.mx, (salvador.raggi, aldrin.barreto, francisco.portillo)@correo.buap.mx

Abstract. Este artículo presenta un nuevo algoritmo de clasificación de patrones basado en kNN con aprendizaje evolutivo. A diferencia de los algoritmos de reducción de prototipos, en donde se suelen eliminar muestras apartadas de las fronteras de clasificación, y hacer sobrevivir muestras originales ubicadas en dichas fronteras, el algoritmo propuesto aquí, mejora gradualmente las posiciones de un conjunto creciente de prototipos sintéticos, cuyo tamaño aumenta a una tasa inferior a la cantidad de muestras de muestras utilizadas. Los resultados experimentales demuestran que el desempeño en clasificación, con el número de prototipos sintéticos creados, es mucho mejor que realizar una selección aleatoria de un número idéntico de muestras, y utilizarlas como prototipos en un kNN tradicional. Así mismo, también se demuestra que la exactitud de clasificación de nuestro algoritmo es aproximadamente igual, y en algunos casos es mejor, que los métodos actuales de reducción de prototipos.

Keywords: KNN, Prototype Reduction, Gaussian Weighted KNN, Incremental Learning, Evolutive Learning.

1 Introducción

El algoritmo *kNN* es uno de los métodos de clasificación supervisados más utilizados gracias a que presenta buenos resultados, es fácil de entender e implementar, sin embargo, los requerimientos de almacenaje y el costo computacional aumentan a medida que el número de muestras incrementa, esto se debe a que la muestra a clasificar debe de ser comparada con cada una de las muestras almacenadas.

Se han presentado diversas soluciones para enfrentar este problema mediante la selección de prototipos representativos del conjunto de muestras disponibles (selección de prototipos) y mediante la creación de nuevos prototipos aplicando múltiples operaciones sobre el conjunto de datos original (generación de prototipos) [1].

Las mejoras que se han planteado se enfocan en aspectos como: la reducción de instancias, el mejoramiento de la exactitud en la clasificación y en la aceleración del

Bibliografía

- [1] T. J. M. Cleophas and A. H. Zwinderman, *Machine learning in medicine.*, vol. 2. Springer, 2013.
- [2] F. Hiroshi, “Ai-based computer-aided diagnosis (ai-cad): the latest review to read first,” *Radiological Physics and Technology*, 2020.
- [3] C.-M. Chen, Y.-H. Chou, N. Tagawa, and Y. Do, “Computer-aided detection and diagnosis in medical imaging,” *Computational and mathematical methods in medicine*, vol. 2013, p. 790608, 09 2013.
- [4] M. Niederman, “Community-acquired pneumonia,” *Annals of Internal Medicine*, 2009.
- [5] W. H. Organization, “Neumonía.” url<https://www.who.int/es/news-room/fact-sheets/detail/pneumonia>, 2019. Accessed: 09- Feb- 2020.
- [6] H. P. MEXICO, “World life expectancy.” url<https://www.worldlifeexpectancy.com/country-health-profile/mexico>, 2017. Accessed 18 Feb. 2020.
- [7] W. A.-M. e. a. S. A. Alzahrani, M. A. Al-Salamah, “Systematic review and meta-analysis for the use of ultrasound versus radiology in diagnosing of pneumonia,” *Critical Ultrasound J.*, vol. 9, no. 6, 2017.
- [8] Y. Amatya, J. Rupp, F. M. Russell, J. Saunders, B. Bales, and D. R. House, “Diagnostic use of lung ultrasound compared to chest radiograph for suspected pneumonia in a resource-limited setting,” *International Journal of Emergency Medicine*, vol. 11, no. 1, 2018.
- [9] A. Ticinesi, F. Lauretani, A. Nouvenne, G. Mori, G. Chiussi, M. Maggio, and T. Meschi, “Lung ultrasound and chest x-ray for detecting pneumonia in an acute geriatric ward,” *Medicine*, vol. 95, no. 27, 2016.

- [10] A. B. Moberg, U. Taléus, P. Garvin, S.-G. Fransson, and M. Falk, “Community-acquired pneumonia in primary care: Clinical assessment and the usability of chest radiography,” *Scandinavian Journal of Primary Health Care*, vol. 34, no. 1, 2016.
- [11] K. S. Saladin, *ANATOMÍA Y FISIOLOGÍA. La unidad entre forma y función*. McGRAW-HILL, 6 ed., 2012.
- [12] F. Netter, *Atlas de Anatomía Humana*. Elsevier, 6 ed., 2015.
- [13] C. A. Asenjo and R. A. Pinto, “Características anatómico-funcional del aparato respiratorio durante la infancia,” *Revista Médica Clínica Las Condes*, vol. 28, no. 1, pp. 7–19, 2017.
- [14] J. A. F. T. et al, *Fisiología humana*, ch. Fisiología del sistema respiratorio, pp. 586–674. McGRAW-HILL, 3 ed., 2005.
- [15] R. Joarder and N. Crundwell, *Chest X-Ray in Clinical Practice*, vol. 23. Springer, 2009.
- [16] J. Soto Campo and V. R. J., *Manual de diagnóstico y terapéutica en neumología*, ch. Radiología de tórax, pp. 13–42. Majadahonda: Ergón., 3 ed., 2016.
- [17] D. Manning, S. Ethell, T. Donovan, and T. Crawford, “How do radiologists do it? the influence of experience and training on searching for chest nodules,” *Radiography*, vol. 12, no. 2, pp. 134 – 142, 2006.
- [18] D. Das and D. C. Howlett, “Chest x-ray manifestations of pneumonia,” *Surgery (Oxford)*, vol. 27, no. 10, pp. 453 – 455, 2009. Infection.
- [19] Y.-W. Kim and L. F. Donnelly, “Round pneumonia: imaging findings in a large series of children,” *Pediatric Radiology*, vol. 37, pp. 1235–1240, 2007.
- [20] R. Sarkar, A. Hazra, K. Sadhu, and P. Ghosh, “A novel method for pneumonia diagnosis from chest x-ray images using deep residual learning with separable convolutional networks,” in *Computer Vision and Machine Intelligence in Medical Image Analysis* (M. Gupta, D. Konar, S. Bhattacharyya, and S. Biswas, eds.), (Singapore), pp. 1–12, Springer Singapore, 2020.
- [21] P. Chhikara, P. Singh, P. Gupta, and T. Bhatia, “Deep convolutional neural network with transfer learning for detecting pneumonia on chest x-rays,” in *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals* (L. C. Jain, M. Virvou, V. Piuri, and V. E. Balas, eds.), (Singapore), pp. 155–168, Springer Singapore, 2020.

- [22] S. Vijendran and R. Dubey, “Deep online sequential extreme learning machines and its application in pneumonia detection,” in *2019 8th International Conference on Industrial Technology and Management (ICITM)*, pp. 311–316, 2019.
- [23] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 839–846, 1998.
- [24] A. Sharma, D. Raju, and S. Ranjan, “Detection of pneumonia clouds in chest x-ray using image processing approach,” in *2017 Nirma University International Conference on Engineering (NUiCONE)*, pp. 1–4, 2017.
- [25] H. Behzadikhormouji, H. Rostami, S. Salehi, T. Derakhshande-Rishehri, M. Masoumi, S. Salemi, A. Keshavarz, A. Gholamrezanezhad, M. Assadi, and A. Batouli, “Deep learning, reusable and problem-based architectures for detection of consolidation on chest x-ray images,” *Computer Methods and Programs in Biomedicine*, p. 105162, 10 2019.
- [26] K. ZUIDERVELD, “Contrast limited adaptive histogram equalization,” *Graphics Gems*, pp. 474–485, 1994.
- [27] S. Rakshit, I. Saha, M. Wlasnowolski, U. Maulik, and D. Plewczynski, “Deep learning for detection and localization of thoracic diseases using chest x-ray imagery,” in *Artificial Intelligence and Soft Computing* (L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, eds.), (Cham), pp. 271–282, Springer International Publishing, 2019.
- [28] K. Wang, X. Zhang, S. Huang, and F. Chen, “Automatic detection of pneumonia in chest x-ray images using cooperative convolutional neural networks,” in *Pattern Recognition and Computer Vision* (Z. Lin, L. Wang, J. Yang, G. Shi, T. Tan, N. Zheng, X. Chen, and Y. Zhang, eds.), (Cham), pp. 328–340, Springer International Publishing, 2019.
- [29] T. B. Chandra and K. Verma, “Pneumonia detection on chest x-ray using machine learning paradigm,” in *Proceedings of 3rd International Conference on Computer Vision and Image Processing* (B. B. Chaudhuri, M. Nakagawa, P. Khanna, and S. Kumar, eds.), (Singapore), pp. 21–33, Springer Singapore, 2020.
- [30] A. Khatri, R. Jain, H. Vashista, N. Mittal, P. Ranjan, and R. Janardhanan, *Pneumonia Identification in Chest X-Ray Images Using EMD*, pp. 87–98. Springer, 01 2020.

- [31] M. Toğaçar, B. Ergen, Z. Cömert, and F. Özyurt, “A deep feature learning model for pneumonia detection applying a combination of mrmr feature selection and machine learning models,” *IRBM*, 2019.
- [32] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 3462–3471, IEEE, Jul 2017.
- [33] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, and S. L. e. a. Baxter, “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *Cell*, vol. 172, pp. 1122–1131, 2018.
- [34] D. Kermany, K. Zhang, and M. Goldbaum, “Labeled optical coherence tomography (oct) and chest x-ray images for classification,” 2018.
- [35] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, “Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists’ detection of pulmonary nodules,” *AJR*, pp. 71–74, 2000.
- [36] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, and M. T. Islam, “Can ai help in screening viral and covid-19 pneumonia?,” *IEEE Access*, vol. 8, pp. 132665–132676, 2020.
- [37] I. Sirazitdinov, M. Kholiavchenko, T. Mustafaev, Y. Yixuan, R. Kuleev, and B. Ibragimov, “Deep neural network ensemble for pneumonia localization from a large-scale chest x-ray database,” *Computers and Electrical Engineering*, vol. 78, pp. 388 – 399, 2019.
- [38] G. Liang and L. Zheng, “A transfer learning method with deep residual network for pediatric pneumonia diagnosis,” *Computer methods and programs in biomedicine*, vol. 187, 2020. Cited By :4.
- [39] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. de Albuquerque, “A novel transfer learning based approach for pneumonia detection in chest x-ray images,” *Applied Sciences (Switzerland)*, vol. 10, no. 2, 2020. Cited By :6.

- [40] K. T. Islam., S. Wijewickrema, A. Collins., and S. O’Leary., “A deep transfer learning framework for pneumonia detection from chest x-ray images,” in *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pp. 286–293, INSTICC, SciTePress, 2020.
- [41] A. K. Acharya and R. Satapathy, “A deep learning based approach towards the automatic diagnosis of pneumonia from chest radio-graphs,” *Biomedical and Pharmacology Journal*, vol. 13, no. 1, pp. 449–455, 2020.
- [42] T. Rahman, M. E. H. Chowdhury, A. Khandakar, K. R. Islam, K. F. Islam, Z. B. Mahbub, M. A. Kadir, and S. Kashem, “Transfer learning with deep convolutional neural network (cnn) for pneumonia detection using chest x-ray,” *Applied Sciences (Switzerland)*, vol. 10, no. 9, 2020.
- [43] S. Yao, Y. Chen, X. Tian, R. Jiang, and S. Ma, “An improved algorithm for detecting pneumonia based on yolov3,” *Applied Sciences (Switzerland)*, vol. 10, no. 5, 2020.
- [44] G. Labhane, R. Pansare, S. Maheshwari, R. Tiwari, and A. Shukla, “Detection of pediatric pneumonia from chest x-ray images using cnn and transfer learning,” in *Proceedings of 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things, ICETCE 2020*, pp. 85–92, 2020.
- [45] X. Xie, X. Li, S. Wan, and Y. Gong, *Mining X-Ray Images of SARS Patients*, pp. 282–294. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
- [46] N. Mohd Noor, O. Mohd Rijal, A. Yunus, A. A. Mahayiddin, C. P. Gan, E. L. Ong, and S. A. R. Abu Bakar, *Texture-Based Statistical Detection and Discrimination of Some Respiratory Diseases Using Chest Radiograph*, pp. 75–97. Singapore: Springer Singapore, 2014.
- [47] S. Varela-Santos and P. Melin, *Classification of X-Ray Images for Pneumonia Detection Using Texture Features and Neural Networks*, vol. 862 of *Studies in Computational Intelligence*. Springer, 2020.
- [48] N. Vamsha Deepa, N. Krishna, and G. Hemanth Kumar, “Feature extraction and classification of x-ray lung images using haralick texture features,” in *Smart and Innovative Trends in Next Generation Computing Technologies* (P. Bhattacharyya, H. G.

- Sastry, V. Marriboyina, and R. Sharma, eds.), (Singapore), pp. 899–907, Springer Singapore, 2018.
- [49] A. A. E. Ambita, E. N. V. Boquio, and J. Naval, P. C., “Locally adaptive regression kernels and support vector machines for the detection of pneumonia in chest x-ray images,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12034 LNAI, pp. 129–140, 2020.
- [50] L. L. G. Oliveira, S. A. e Silva, L. H. V. Ribeiro, R. M. de Oliveira, C. J. Coelho, and A. L. S. S. Andrade, “Computer-aided diagnosis in chest radiography for detection of childhood pneumonia,” *International Journal of Medical Informatics*, vol. 77, no. 8, pp. 555 – 564, 2008.
- [51] S. Antani, S. Cademir, S. Jaeger, and A. Karargyris, “Automated detection of lung diseases in chest x-rays,” 2015.
- [52] B. Ginneken, S. Katsuragawa, B. ter Haar Romeny, K. Doi, and M. Viergever, “Automatic detection of abnormalities in chest radiographs using local texture analysis,” *IEEE transactions on medical imaging*, vol. 21, pp. 139–49, 03 2002.
- [53] C. for Artificial Intelligence in Medicine and Imaging, “Rsnas 2017: Rads who use ai will replace rads who don’t.” [urlhttps://aimi.stanford.edu/news/rsna-2017-rads-who-use-ai-will-replace-rads-who-don-t](https://aimi.stanford.edu/news/rsna-2017-rads-who-use-ai-will-replace-rads-who-don-t), 2017. Accessed: 10- Feb- 2020.
- [54] S. J. Russell and N. Peter, *Inteligencia artificial un enfoque moderno*. Pearson Pentice hall, 2 ed., 2004.
- [55] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning*. Cambridge University press, 2014.
- [56] E. Wolfgang, *Introduction to Artificial Intelligence*. Springer, 2 ed., 2017.
- [57] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [58] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 4 ed., 2018.
- [59] R. Larson and D. C. Falvo, *Fundamentos de Álgebra Lineal*. Cengage Learning, 6 ed., 2010.

-
- [60] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.
- [61] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 01 1991.
- [62] D. Mery, *Computer Vision for X-Ray Testing*. Springer, first ed., 2015.
- [63] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 711–720, 07 1997.
- [64] I. Triguero, J. Derrac, S. Garcia, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 86–100, 2012.
- [65] S. Maheshwari, R. R. Sharma, and M. Kumar, "Lbp-based information assisted intelligent system for covid-19 identification," *Computers in biology and medicine*, vol. 134, 2021.